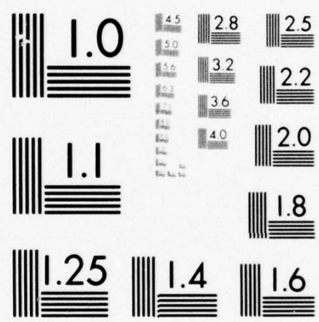
AD-A039 308 SYSTEMS CONSULTANTS INC WASHINGTON D C F/G 17/4
ANALYSIS OF SURFACE IR DECOY PLACEMENT FOR ANTISHIP MISSILE DEF--ETC(U)
MAY 77 S C SCHIFF N00039-76-C-0296
UNCLASSIFIED SCI-6126-006/1 NL

| OF |

AD
A039308

END

DATE
FILMED
5-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

6126-006/1

FG.

1

ANALYSIS OF SURFACE IR DECOY PLACEMENT FOR ANTISHIP MISSILE DEFENSE

VOLUME I: STUDY METHODOLOGY

6 May 1977

Prepared for
COMMANDER, NAVAL ELECTRONIC SYSTEMS COMMAND
PME 107-52

Under Contract Number
N00039-76-C-0296

SYSTEMS CONSULTANTS, INC.
1054 31st. Street, N.W.
Washington, D.C. 20007

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

6 ANALYSIS OF SURFACE IR DECOY PLACEMENT FOR ANTISHIP MISSILE DEFENSE. Volume I. Study Methodology.

5. TYPE OF REPORT & PERIOD COVERED

9 Final Rep[redacted]t. 15 Apr[redacted]76-15 Feb '77

6. PERFORMING ORG. REPORT NUMBER

14 SCI-6126-006/1

7. AUTHOR(s)

10 S.C. Schiff

8. CONTRACT OR GRANT NUMBER(s)

15 N00039-76-C-0296

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Systems Consultants, Inc.
1054 31st. Street, N.W.
Washington, D.C. 20007

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Electronic Systems Command
Washington, D.C. 20360
Attn: PME 107-52

12. REPORT DATE

11 6 May [redacted]77

13. NUMBER OF PAGES

61 12 60p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

V2 C009960

18. SUPPLEMENTARY NOTES

388 950

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Infrared Countermeasures
Antiship Missile Defense
Decoys
Digital Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The placement efficacy of surface-borne infrared decoy systems for countering antiship missiles employing the pushbroom scan mechanism have been analysed by means of digital simulation techniques. The study report has been published in two parts; this one describes the simulations in terms of mathematical models, evaluation criteria and software. The second part of the report includes the results of an analysis for six missile variants.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

(front master) 1 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

I. INTRODUCTION

This report describes the methodology for and results of an analysis of ship defense infrared decoy placement conducted by Systems Consultants, Inc. (SCI) for the Naval Electronic Systems Command under Contract N00039-76-C-0296. The purpose of the contract was to extend earlier work (performed by SCI for the Naval Surface Weapon Center White Oak Laboratories under Contract N60921-75-C-0237) to surface-borne decoys and to consider the effects of missile parameters variations upon decoy placement effectiveness. A secondary purpose of this report is to more completely document the analysis methodology used in evaluating decoy placement effectiveness.

In the interest of achieving widespread dissemination of the analysis methodology, this report is divided into two volumes. This volume deals exclusively with the analysis tools used in evaluating decoy placement effectiveness. The second volume, which is classified Confidential, contains the results of application of the simulations for six missile variants.

II. OVERVIEW OF THE SIMULATIONS

A. BACKGROUND

A decoy is a device which is designed to present an appearance which is indistinguishable from that of a true target when viewed by a specified sensor, so that deception of the sensor or its operator can be accomplished. The degree to which the decoy must resemble the true target, in terms of all of its measurable characteristics, depends upon the attributes of the sensor. For example, a sensor which measures both the signature and motion characteristics of its target cannot be reliably deceived by a decoy which differs significantly from a true target in either respect.

Decoys have been in use since the earliest days of radar; the advent of infrared (IR) targeting and guidance systems has resulted in application of decoys against IR sensors as well as radars. The ways in which decoys can be used to defeat weapons correspond to the sequence of operations of weapon targeting. During World War II, for example, chaff decoys were used by the allies against the German air defenses in two ways: corridor chaff was employed to mask entire flights of bombers to deny information on the number of aircraft involved; chaff was also used to degrade the tracking capabilities of the German gun-laying radars. Currently, decoys are being developed to counter antiship missiles (ASMs). Against these, there can be as many as three modes of decoy operation:

(1) Target Selection Confusion of the Operator (TSCO), wherein the purpose of the decoy is to cause the weapon to be launched against a decoy rather than a true target or to deny targeting data.

(2) Initial Acquisition Decoying. Many guided weapons, particularly long range ones, do not track their targets continuously from launch, but fly under the control of an autopilot for some specified period, after which the target must be acquired. The purpose of initial acquisition (IA) decoying is to cause the missile to acquire a decoy rather than a true target.

(3) Break Lock Decoying. Once a missile has acquired a true target, break lock (BL) decoys can be used to degrade the tracking accuracy of the missile.

The foregoing example thus represents instances of TSCO and BL decoying.

The efficacy of a decoy depends upon satisfaction of two criteria: indistinguishability from a true target and placement relative to the true target and the sensor. The decoy must be indistinguishable from a true target in the sense that the victim sensor cannot reject the decoy on the basis of received data. The decoy's position is important because sensors accept only those signals which emanate from the volume of space where true targets are expected to be. Sensor characteristics may differ according to their mode of operation (pre-launch targeting, acquisition of the target by the weapon, and terminal homing) and so the requirements for decoy characteristics and placement may also vary according to decoy mode (TSCO, IA or BL). Since both the indistinguishability and placement criteria must be satisfied for the decoy to be effective, they can be evaluated independently in most instances. This principle is central to the utility of the two digital simulations described in this report, which are used to evaluate decoy placement in the IA and BL modes against IR guided ASM's employing the pushbroom scan mechanism.

The decoy placement criterion can always be satisfied if there is sufficient selectability in decoy placement relative to the defended ship (true target) and the victim, and if the timing of the decoy launch is appropriate. However, practical limitations to decoy placement will always exist. Some of these limitations arise as a result of uncertainty as to the location (range,altitude and direction), speed, direction of motion and detailed characteristics of the ASM. Others result from the design limitations of the decoy system: in order to minimize equipment cost and size, shipboard decoy launchers employ fixed train and elevation angles; the decoys themselves are ballistic and use a single propulsion system. Consequently, the design of decoy systems to achieve

maximum placement effectiveness and the evaluation of the placement effectiveness of a decoy system are relatively complicated problems. The simulations described subsequently in this report were developed specifically to address these problems.

B. SCOPE

Decoy effectiveness against a particular ASM depends upon decoy the placement geometry and also upon the performance characteristics of the decoy (e.g., its signature). Consequently, it is possible to determine decoy effectiveness only through consideration of both performance and placement. Since these are independent of each other for most applications (and in particular for the threat model used herein) it is possible to write the following expression for decoy effectiveness:

$$\eta = \eta_{placement} \eta_{performance}$$

where $\eta_{placement}$ is the Placement effectiveness and $\eta_{performance}$ is the Performance effectiveness, i.e. the probability that the missile will home on the decoy given that the decoy is placed within its field of view. The decoy placement simulations have been designed to calculate $\eta_{placement}$, and therefore yield η, the decoy effectiveness, only under the condition that $\eta_{performance}$ is equal to unity. That is, the simulations give the upper bound on decoy effectiveness only.

The results from the simulation are further limited because of designed-in assumptions about the nature of the engagement. Placement effectiveness depends, inter-alia, upon the range and direction of the threat at the time of decoy launch, the speed of the ship, and the speed and direction of the true wind (these last two factors apply only to decoys which remain airborne and thus drift with the wind). The simulations yield average placement effectiveness values for a given ship speed,decoy launcher, and decoy against a specified threat by sampling from a four dimensional space which is bounded by minimum and maximum threat directions, threat ranges, wind directions and wind speeds. The first three of these are uniformly distributed, i.e. any value between

the bounds is equally likely. The wind speed is not uniformly distributed, but is chosen from a parent population selected to closely match empirical data (Long. et al, 1965). In the IA simulation, the sample space includes two additional dimensions so that normally distributed missile targeting errors (range and angle) can be taken into account. Thus the average results of the simulation cannot be applied to any one scenario, but represent the placement effectiveness achieved over a large number of scenarios which occur under varying conditions. Since both simulations have provisions for detailed reports giving the placement results on a sample by sample basis, they can also be used in evaluating decoy placement performance in any particular scenario.

C. STRUCTURE

Both simulations represent embodiments of mathematical models and evaluation criteria in software. There are three elements to each mathematical model: the decoy, ship and threat. The mathematical model describes the salient features of each element and their motion relative to each other. The evaluation criteria consist of sets of precise conditional statements, the outcomes of which are classified as pass or fail with respect to various success determinants. Implemented in software, the mathematical model and evaluation criteria are used with statistically independent samples of the appropriate variables to perform a time line analysis of an engagement. By manifold replication of this process, average placement effectiveness values are obtained (Monte Carlo method). The software also includes report generators of two types: summary and detailed. The summary reports yield aggregated results (e.g. average placement effectiveness) while the optional detailed reports yield results (values of the random variables and pass/fail indication) on a case by case basis.

The mathematical models upon which the simulations are based are discussed in Section III of this report. Section IV describes the evaluation criteria, and Sections V and VI include FORTRAN IV software and interactive user's guides for the BL and IA simulation programs.

III. THE MODEL

Both the initial acquisition simulation (IADS.ONE) and the break lock simulation (IRDPS.F4) include sets of equations for the coordinates of the ship, decoy and threat in two reference frames: the earth frame (so called because it is at rest on the surface of the earth) and the missile frame (which is at rest with respect to the missile). These reference frames are connected by means of a coordinate transformation; at any instant the transformation consists of a translation plus a rotation of coordinates. Once the transformation is effected, the coordinates of any point are given in the missile frame. By referencing the missile frame to its line of flight direction, the angles from boresight to any point in space can be determined, that is, it is determinable whether any point lies within the missile seeker field of view. Naturally, it is also easy to determine the angle subtended by an object, and also distances. In the following subsections the equations of motion for the ship, decoy and missile are developed. Field of view considerations are discussed in section IV, as they are naturally related to the evaluation criteria.

A. SHIP

For the purposes of these simulations the ship is represented as a line of length 2BSHD. Its motion is uniform in the earth frame, that is, it travels at a constant speed in a constant direction. The reference direction is chosen to be along the x-axis of the earth frame; at time T=0 the midpoint of the ship is located at the origin of the coordinate system. The equations of motion for the ship's midpoint are thus:

$$X_S = V_S T$$

$$Y_S = Z_S = 0$$

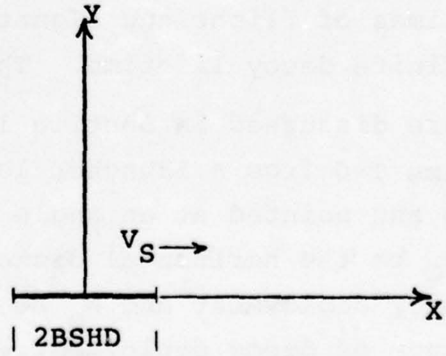


Figure 1

while the coordinates of the bow and stern are shifted along the x-axis by +BSHD and -BSHD, respectively:

$$X_{SB}=X_S+BSHD \qquad (Y_{SB}=Z_{SB}=0) \qquad (2)$$

$$X_{SS}=X_S-BSHD \qquad (Y_{SS}=Z_{SS}=0)$$

It has been determined that the ship can be accurately modeled in terms of a line rather than a three dimensional body for the following reasons:

(1) The missile seeks the interface between the ship and the ocean, so that it is not necessary to represent the ship in terms of a vertical extent.

(2) The ship is very long in comparison to its width (typical destroyers have L/B ratios of about 9), so as a consequence the width of a ship contributes little to its angular extent when viewed from distances of practical interest. Further, at these distances the angle subtended by the ship's width (when viewed bow-on or from astern) is typically smaller than the resolution capability of the missile. Thus it is not necessary to complicate the model by treating the ship as even a two dimensional body.

B. DECOY

A decoy is treated within the model as an object which in effect comes into existence at some calculated time and which ceases to exist at a later time. This reflects the need to model the times of flight and signature development of the decoy, and the finite decoy lifetime. These times are called T_f, T_b and T_{life} and are discussed in Section IV. Suppose that the decoy is launched at time T=0 from a launcher located a distance d_L forward of amidships and pointed at an angle ϕ off the bow, as shown in Figure 2. Let R_C be the horizontal distance of decoy deployment and h_o be the altitude of decoy deployment. Since the decoy initially has a velocity component V_S in the X direction, its position at time $T=T_f$ will be given by

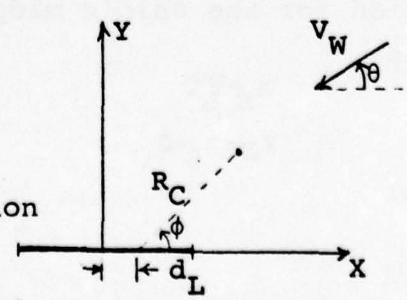


Figure 2

$$X_C = d_L + R_C\cos\phi + V_S T_f$$
$$Y_C = R_C\sin\phi \qquad (3)$$
$$Z_C = h_o$$

If the decoy is surface borne ($Z_C = 0$) then these coordinates give the position of the decoy in the earth frame for all later times, since ocean currents are neglected in the model. However, if the decoy is airborne, it will be subject to vertical motion under the influence of gravity and horizontal motion under the influence of the true wind, as measured in the earth frame. Let θ be the angle of the true wind as measured off the x-axis in Figure 2 and let the true wind speed be V_W. Suppose also that the decoy quickly reaches its terminal velocity of fall V_{CV}. If in addition the decoy is launched at time $T=T_L$, the coordinates of the decoy at time $T>T_f+T_L$ are

$$X_C = d_L + R_C\cos\phi + V_S(T_f+T_L) - V_W(T-T_L)\cos\theta$$
$$Y_C = R_C\sin\phi - V_W(T-T_L)\sin\theta \qquad (4)$$
$$Z_C = h_o - V_{CV}(T-T_f-T_L)$$

The break lock simulation program differs from the initial acquisition simulation program in that T_L is set to zero. The simulations also differ in that the decoy is treated as a vertical line in the BL simulation (IRDPS.F4) while in the initial acquisition program IADS.ONE it is treated as a point. These representations have been found to be adequate based upon experience. See Schiff and Pieklo (1976).

If the decoy is airborne, its horizontial motion relative to the ship will be determined by the relative wind (wind speed and direction as measured by the ship). Since the relative wind conditions are of tactical importance, the calculation of relative wind speed and direction are included in the simulation. The relative wind speed is given by

$$V_{RW} = \sqrt{V_S^2 + V_W^2 + 2V_S V_W\cos\theta}$$

while the direction is

$$RWA = \sin^{-1}(V_W\sin(\theta)/V_{RW})$$

This second equation has two admissible roots which differ by 180^{o} - 2RWA. The physically correct root is determined by considering which quadrant the true wind lies in and its speed relative to that of the ship.

C. MISSILE

Although the decoy and ship models used in the analysis of decoy placement can be made quite general, such is not possible for the missile, because of the wide range in possibilities for IR-guided ASM design. Consequently, the model discussed herein is in general representative of but one generic family of ASMs. This generic family is called the "pushbroom" type because of the analogy between its scanned field of view (rectangular in cross section with its long dimenstion orghogonal to the line of flight, and fixed relative to the missile, thereby achieving its search in the direction of flight due to its motion) and the swept path of a pushbroom. The additional general characteristics of the seeker model are:

(1) Fixed flight altitude until the target-ocean interface reaches the center of the elevation of view, whereupon a constant slope dive is undertaken to maintain the elevation track point at the target water line.

(2) Fixed ground speed.

(3) Adaptive azimuth gate in the track mode to include the edges of the target within the FOV, permitting location of its angles.

(4) Proportional navigation in the azimuth plane.

Practically speaking, the model is limited in terms of its validity to these pushbroom seekers which acquire their targets at ranges of about five nautical miles or less, because at ranges significantly greater than this there is the potential for propogation effects to preclude target detection or accurate tracking even though the target lies within the field of view. If a particularly strong target and decoy, or excellent atmospheric conditions, are

assumed, then the model remains valid for somewhat larger ranges, until the flat earth approximation used in calculating angular relationships begins to fail. The degree of accuracy of the flat-earth approximation depends upon the down-look angle to the top of the seeker field of view; if the horizon is included in the field of view of the seeker the flat-earth approximation cannot be used even at short ranges. Consequently, the pushbroom model assumes that the sightline depression angle to the top of the elevation field of view is always positive.

For reasons of economy of simulation design and execution, the flight of the missile is decomposed into two segments: prior to and subsequent to detection of the decoy. From its initial point the missile moves along a straight line toward the estimated point of impact with the ship midpoint until the decoy is detected. Upon satisfaction of the decoy detection criterion, the missile turns toward the decoy at a rate which is proportional to the azimuth angle off boresight to the decoy, and descends as appropriate. Should the decoy lose its viability prior to the arrival of the missile, or if it is possible for the missile to fly through the decoy and remain airborne, the missile continues in the direction of flight at the time of decoy extinction or fly-through, and re-enters the search mode (IA program).

The two segments of the missile dynamic model represent different navigational modes: lead and proportional. This is in contrast to real missiles, which would employ a single navigational mode. However, the differences betwen the flight paths for the two modes depend in the main upon the relationship between the target and missile velocities. Since the missile speed is at least an order of magnitude greater than those of either the ship or the decoy, the difference between the flight profiles for the two modes is insignificant, and thus the simplification made is the model is entirely justified.

The initial phase of missile motion is described identically in both simulation models, with one difference in detail: The IA simulation allows for targeting errors while the BL simulation does not. At time $T=0$ the missile is a distance R_o

from the ship at bearing γ. The missile is moving in the xy plane with speed V_T, while the ship moves along the x-axis of the earth frame with speed V_S. Let T_e be the time of impact; the missile will move a distance $V_T T_e$ and the ship will move a distance $V_S T_e$. Using the law of sines,

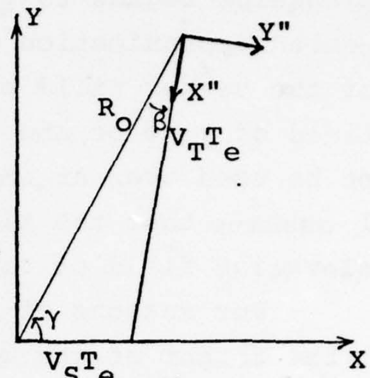


Figure 3

$$\beta = \sin^{-1}\left\{\frac{V_S \sin\gamma}{V_T}\right\} \qquad (5)$$

subject to $V_S \leq V_T$, and

$$T_e = \frac{R_o \sin\gamma}{V_T \sin(\beta+\gamma)} \qquad (6)$$

With these relationships the motion of the missile in the xy plane is given by

$$X_T = R_o \cos\gamma\,(1 - T/T_e) + V_S T$$

$$Y_T = R_o \sin\gamma\,(1 - T/T_e)$$

for times $T \leq T_e$.

The missile frame of reference is defined as that which is at rest with respect to the missile, with its origin corresponding to the missile location and its x-axis directed along the missile line of flight, as shown in Figure 3. The transformation between the earth frame and the missile frame is accomplished by first making a translation

$$X' = X - X_T$$

$$Y' = Y - Y_T$$

followed by a rotation in a counter clockwise direction

$$X'' = -X'\cos(\beta+\gamma) - Y''\sin(\beta+\gamma)$$

$$Y'' = X'\sin(\beta+\gamma) - Y'\cos(\beta+\gamma)$$

or

$$X'' = (X_T - X)\cos(\beta+\gamma) + (Y_T - Y)\sin(\beta+\gamma) \qquad (7)$$

$$Y'' = (X - X_T)\sin(\beta+\gamma) + (Y_T - Y)\cos(\beta+\gamma)$$

The coordinate transformation also includes a translation in the vertical direction:

$$Z'' = Z - Z_T \qquad (8)$$

The above set of transformation equations can be written in a simpler form provided the missile is in uniform rectilinear motion. In this event, the position of an object in the missile frame is given in terms of the initial position of the missile in the earth frame and the equations of motion for other objects moving relative to the earth frame. These equations, used in the IA simulation program, are

$$X'' = (R_O\cos\gamma - X)\cos(\beta+\gamma) + (R_O\sin\gamma - Y)\sin(\beta+\gamma) - V_T T$$

$$Y'' = (X - R_O\cos\gamma)\sin(\beta+\gamma) + (R_O\sin\gamma - Y)\cos(\beta+\gamma) \qquad (9)$$

$$Z'' = Z - Z_T$$

In the foregoing, it was assumed that the missile was perfectly targeted. While this is a reasonable assumption to make for the purposes of break lock decoy placement analysis, it is not justifiable for the general initial acquisition case. Consequently, the initial acquisition simulation includes a method for describing targeting errors. This is accomplished in the following manner: Let R_O, γ and β be defined as before; let ε be the angle targeting error and let δR be the range targeting error, as shown in Figure 4. In the absence of errors, the missile would impact the ship at P at time $T_e = |TP|/V_T$. In the presence of the errors ε and R, the predicted point of impact is Q, the predicted time of impact is $|TQ|/V_T$, and the time at which the missile crosses the ship track is $|TR|/V_T$. The former of these quantities is required to determine the timing of initiation of the missile target acquisition process, while the latter is used in the simulation to terminate an individual case (the missile cannot acquire the ship once it has crossed the x-axis).

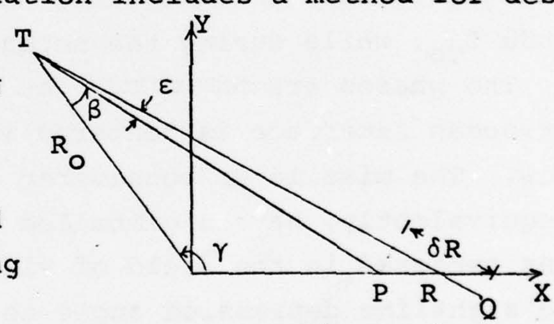


Figure 4

The length of the segment TR in Figure 4 is determined using the law of sines:

$$|TR| = \frac{R_O\sin\gamma}{\sin(\beta+\gamma+\varepsilon)} \qquad (10)$$

where β is given by equation (5). Thus the time at which the missile will cross the x-axis is

$$T_e = \frac{-R_o \sin\gamma}{V_T \sin(\beta+\gamma+\varepsilon)} \tag{11}$$

Equations(9) can be modified to reflect targeting errors through the substitution $\beta \rightarrow \beta+\varepsilon$.

Upon satisfaction of target detection criteria or target selection, the missile enters its track mode. In this mode, the path of the missile deviates from rectilinear motion in a plane in two respects: it turns in the horizontal plane, and decends in the vertical plane. The motion in these planes is uncoupled, that is, the motion in each plane is uniquely determined by geometric quantities (angles) measured in that plane.

The flight of the missile in a vertical plane is characterized by two phases. During the first phase of flight, (prior to target detection or selection) the missile flies at a constant altitude Z_{TO}, while during the second phase it dives at a constant rate. The phases are demarcated at the point in space where the target-ocean interface is centered in the missile elevation field of view. The missile is considered to maintain a constant attitude (or, equivalently, have a gimballed sensor) so that the target remains centered in the field of view as the missile dives. Let δ_u be the sightline depression angle to the top of the missile field of view, and let δ_L be the depression angle to the bottom of the field of view. Then the midpoint of the field of view, and consequently the dive angle, is

$$\delta = (\delta_u + \delta_L)/2$$

Consequently, the altitude of the missile is

$$Z_T = \begin{cases} Z_{To}, & \rho \geq Z_{To}\cot\delta \\ \rho\tan\delta, & \rho \leq Z_{To}\cot\delta \end{cases} \tag{12}$$

where $\rho = \sqrt{X''^2 + Y''^2}$ is the horizontal range to the target.

During the tracking segment of the missile flight, the missile trajectory in the horizontal plane is determined by the position of the target relative to the missile line of flight. In particular, the direction and rate of turn in the azimuth plane are related to the angle off boresight by

$$\frac{d\alpha}{dT} = \dot{\alpha} = -\kappa\alpha \tag{13}$$

where $\dot{\alpha}$ is the rate of turn, α is the azimuth angle off boresight, and κ is the navigation system constant. Equation (13) describes a simple proportional navigation system. The constant κ is related to the acceleration limit of the airframe, and in turn to the sensor field of view, as shown below.

Suppose the missile is located at point A as shown in Figure 5; at that instant the missile velocity is directed along the line segment AB; the speed is V_T. It is easy to show that the maximum maneuver requirement (in terms of the missile acceleration) results from detection of a target at P, given that the field of view is the shaded region in the figure. Of all paths which the missile could fly from A to P, that which minimizes the acceleration is one of constant acceleration; this implies that the flight path is an arc of a circle passing through points A and P, constructed so that the initial velocity vector is tangent to the circle. By means of an elementary geometrical construction, the central angle ψ is determined to be

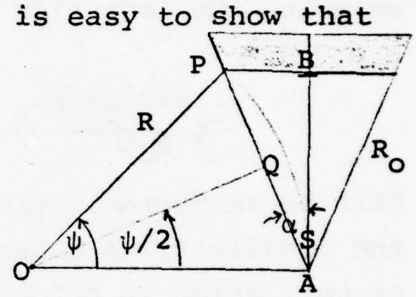


Figure 5

$$\alpha_S = \psi/2.$$

The acceleration of the missile in flying along the arc AP is

$$a = \frac{V_T}{R}$$

Since $AP = R_O$ and Q besects AP, we have

$$\sin(\psi/2) = \frac{|AQ|}{|0A|} = \frac{Ro}{2R}$$

so that

$$R = R_O / 2\sin\alpha_S$$

and hence

$$a=\frac{2V^2{}_T\sin\alpha_S}{Ro} \qquad (14)$$

gives the maximum acceleration required in terms of the missile speed and field of view. Note that a motionless target is assumed; for the present application it is reiterated that V_T typically exceeds the target speeds by about two orders of magnitude so that the correction required to accommodate target motion is small. If we write the left side of (14) in terms of the missile "g" limit L_g,

$$L_g=\frac{2V^2{}_T\sin\alpha_S}{R_o g} \qquad (15)$$

This is an approximation because it is based on the assumption that the missile flies along an arc of constant radius for a motionless target, which is not precisely true for a proportional navigation system.

To relate the maneuver limit L_g to the navigational system constant κ (kappa), the acceleration is determined by calculating the velocity at times T and $T+\delta T$ where a translation $V_T\delta T$ is performed at the beginning of each time step followed by a rotation $\kappa\alpha\delta T$ at the conclusion. Then, using the formal definition of a derivative, the acceleration is obtained. Kappa is then found by solving the resulting equation for the case $a=gL_g$ and $\alpha=\alpha_S$:

$$\kappa = \frac{gL_g}{V_T\alpha_S} \qquad (16)$$

To represent the maneuvers of the missile in the horizontal plane its coordinates are rotated then translated at each time increment. Let α denote the angle of the target as measured off the missile line flight at time T, and let δT be the time increment used in stepping the motion. Defining γ' to be the polar (azimuth) angle of the missile in the earth frame ($\gamma'=\gamma$ at time T=0) and β' to be the angle between the missile

radius vector and its line of flight ($\beta = \beta'$ at time T=o), the rotated and translated coordinates of the missile in the earth frame are

$$x_T = x_T + V_T \delta T \cos(\gamma' + \beta + \kappa\alpha\,\delta T) \qquad (17)$$

$$y_T = y_T - V_T \delta T \sin(\gamma' + \beta^1 + \kappa\alpha\,\delta T)$$

In addition to the transformation given by (17) the transformation

$$Z' = Z - Z_T$$

is performed where Z_T is given by (12).

Since it is not inconceivable that the target selected for tracking is an airborne decoy, the model must include a provision for the case where the missile impacts a decoy but remains airborne, since a decoy is a soft object and thus cannot be assumed to cause detonation of the missile warhead. The limited maneuverability of the missile can result in its impacting the surface even though the decoy altitude is positive. The condition for this is derived by consideration of Figure 6. In accordance with equations (12), the missile descends at an angle $\frac{\delta_u + \delta_L}{2} = \Delta$.

Upon encountering the decoy (at altitude Z_c) the missile continues to descend for a time δT in accordance with its finite reaction time. It then executes a maximum "g" pullout. Let Z be the missile altitude at the instant it initiates its pullout; it will move along an arc of radius

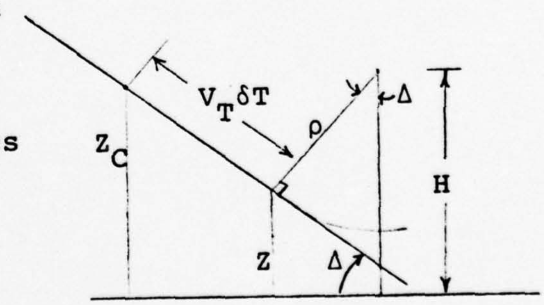


Figure 6

$$\rho = \frac{V^2_T}{L_g g}$$

The missile will be able to effect a recovery if $H > \rho$ But

$$H = Z + \rho\cos\left(\frac{\delta u + \delta L}{2}\right)$$

Consequently, if

$$Z_C - V_T\delta T\sin\left(\frac{\delta u+\delta L}{2}\right) + \frac{V_T^2}{gL_g}\cos\left(\frac{\delta u+\delta L}{2}\right) > \frac{V_T^2}{gL_g}$$

the missile will recover prior to impacting the ocean. Once the missile enters the tracking segment of its flight, the time to impact is calculated. This time of flight is then used to calculate Z_C at the time of impact. If at that time

$$Z_C \leq \left(1-\cos\left(\frac{\delta u+\delta L}{2}\right)\right)V_T^2/gL_g + V_T\delta T\sin\left(\frac{\delta u+\delta L}{2}\right) \tag{18}$$

the missile will impact the sea; otherwise it recovers and continues its flight.

IV. EVALUATION CRITERIA

Implementation in software of the models permits calculation of the trajectories of the ship, decoy and missile under a wide range of initial conditions. This in turn provides the information required to determine the relative positions of the objects at any time, and thus to evaluate decoy placement effectiveness. This section describes the methodology for that evaluation. Since the possibilities for decoy effectiveness differ according to the mode of operation (BL or IA), the evaluation criteria for BL and IA effectiveness are correspondingly different. Consequently, the evaluation criteria for break lock and initial acquisition are treated separately in this section.

A. BREAK LOCK EVALUATION CRITERIA

The BL simulation program IRDPS.F4 evaluates two aspects of decoy placement: whether the decoy is within the field of view of the missile for a sufficient time to affect it (given that the missile is aimed at the ship), and assuming that this is so, whether the decoy will cause a sufficient error in the missile to cause it to miss the ship. It is emphasized from the start that the second of these criteria cannot be as reliable as the first, because it is based on the assumption that track is shifted to the decoy (and not, for example,to the ship-decoy centroid) at some specific time. Nonetheless, the second placement criterion provides an indication of the efficacy of the decoy in causing angle tracking errors.

The first success criterion, called SUCCESS 1 (S_1 for brevity) is based on the purely geometric condition that the decoy be within the missile field of view for a period of time equal to or greater than the detection time of the missile. The detection time of an IR-guided missile is equal to the time required to complete some specified number of scans,N_S. Thus, if the scan frequency is Ω, the detection time is

$$T_D=N_S/\Omega$$

In the simulation IRDPS, time is stepped in fixed increments of δT which may or may not be equal to T_D at the option of the user. To relate δT to T_D, a parameter called ITS is defined as the smallest

integer satisfying the following relationship:

$$ITS > \frac{T_D}{\delta T}$$

Then, for example, if $T_D = \delta T$, ITS=2, so that the decoy must be in the field of view at two successive steps in time to count as a success with respect to S_1. The implementation of this condition is shown in Figure 7, which indicates the structure for determining the outcome of a single engagement in the BL simulation.

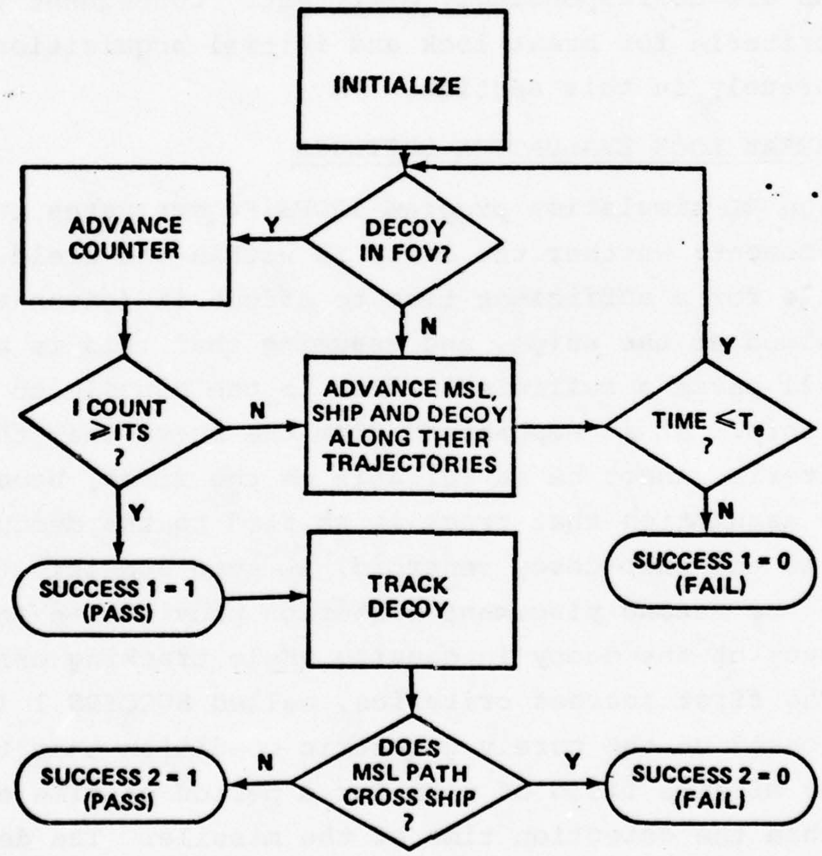


Figure 7

Determination of decoy presence within the missile field of view (FOV) is accomplished by evaluating two angles referenced to the missile: the azimuth angle from the ship midpoint to that of

the decoy, and the depression angle to the decoy. If the ship coordinates in the missile frame are $(X_S", Y_S", Z_S")$ while those of the decoy are $(X_D", Y_D", Z_D")$, then the azimuth angle α_{DS} is found be solving the triangle of Figure 8 for α_{DS} using the law of cosines:

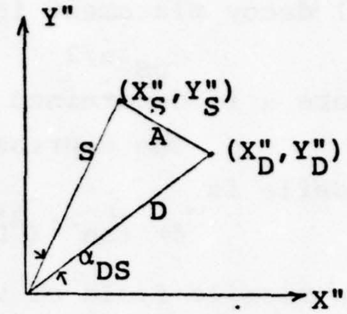


Figure 8

$$\alpha_{DS} = \cos^{-1}\left\{\frac{S^2 + D^2 - A}{2SD}\right\} \quad (19)$$

where

$$S^2 = X_S"^2 + Y_S"^2$$
$$D^2 = X_D"^2 + Y_D"^2$$
$$A^2 = (X_S" - X_D")^2 + (Y_S" - Y_D")^2$$

The value of α_{DS} calculated using equation (19) is compared with the azimuth field α of view of the missile. The missile azimuth field of view is considered to be adaptive, and is determined based upon the azimuth angle subtended by the ship. If α_o is the instantaneous FOV of the missile seeker detector (or of a single detector in the case where the seeker employs a uniform detector array) and α_S is the maximum scanned azimuth FOV of the seeker, then the azimuth FOV of the missile will be

$$\alpha = \alpha_o + 2\sigma \quad (20)$$

in the case where the angle subtended by the ship is smaller than α_o, and

$$\alpha = 2\alpha_S \quad (21)$$

when the angle subtended by the ship is greater than $2\alpha_S$. In equation (20), the quantity σ is introduced to account for the seeker's scan past the ship in the track mode which is required to provide a reference signal and to accommodate rapid maneuvers. In equation (21), the factor 2 is introduced because α_S is measured from boresight.

In the usual case where the azimuth angle ψ subtended by the ship is less than $2\alpha_S$ but greater than α_o the azimuth field of view of the seeker is determined using

$$\alpha = 0.8\psi + 2\sigma \quad (22)$$

where ψ is calculated using (19), except that the coordinates used are those of the ship's bow and stern. Given that the ship is centered in the missile field of view, the requirement for successful decoy placement in azimuth is

$$\alpha_{DS} < \alpha/2 \tag{23}$$

where α is determined using (20), (21), or (22) as appropriate.

The depression angle of the decoy as measured by the missile is

$$\delta = \tan^{-1}(Z''_D / \sqrt{X''^2_D + Y''^2_D})$$

The missile field of view extends from δ_u (top of elevation FOV) to δ_L (bottom of elevation FOV); consequently the requirement for successful decoy placement in elevation is

$$\delta_u < \delta < \delta_L \tag{24}$$

In summary: S_1 is true provided that equations (23) and (24) are simultaneously satisfied for ITS successive evaluations during the viable lifetime of the decoy.

Given that the decoy satisfies S_1, the missile is assumed to immediately initiate maneuver toward the decoy. If the decoy is deployed at a sufficiently low altitude, the missile will impact the sea in the vicinity of the decoy. Otherwise, the missile will continue to fly after impacting the decoy and, in reality though not in the simulation, re-enter its search mode. If the flight of the missile after decoy impact, projected in the direction of flight immediately prior to decoy impact, intersects any part of the ship, it is assumed that the missile will hit the ship. This is evaluated by checking the signs of the Y" coordinates of the ship's bow and stern. If they are both of the same sign, i.e.

$$Y''_{SB} > 0 \text{ and } Y''_{SS} > 0$$
$$\text{or} \tag{25}$$
$$Y''_{SB} < 0 \text{ and } Y''_{SS} < 0$$

then the second success criterion SUCCESS 2 (or S_2) is satisfied.

In summary: S_2 is true provided that either equation (18) or equation (25) holds at the time the missile impacts the decoy.

B. INITIAL ACQUISITION EVALUATION CRITERIA

To treat the initial acquisition mode, it is assumed that at the time of simulation initialization neither the decoy nor the ship lies within the search field of view of the missile. Accordingly, there are three possibilities:

(1) The first target acquired will be the decoy.

(2) The first target acquired will be the ship.

(3) The decoy and ship will be acquired simultaneously.

There is another logical possibility: that neither the ship nor the decoy will be acquired. However, in the realistic situation where modest targeting errors are present, this will not occur.

That the missile detects the decoy prior to detecting the ship does not necessarily imply success in IA decoying, for two reasons. First, owing to the finite lifetime of the decoy and the relatively large distance which the missile must traverse from the time of initialization until impact, there is an appreciable likelyhood that the decoy will cease to be viable before the missile reaches it. Should this occur,the missile will reinitiate the acquisition process. Secondly, if the decoy is at an appreciable altitude at the time the missile impacts it (the decoy being viable at that time) the missile will again reinitiate the acquisition process. These possibilities are accounted for in the simulation IADS.ONE as shown in Figure 9.

In the event that the missile detects the ship prior to detecting the decoy, the decoy is considered to have failed in the initial acquisition mode. It is not necessarily true that the decoy would be useless for break lock given that it failed as an IA decoy, but such would be expected to be the case in the majority of instances.

The final practical possibility, that the decoy and ship are detected simultaneously, is a rarity which is nonetheless considered. Should this condition occur, preferred acquisition of the decoy over the ship is assumed to take place.

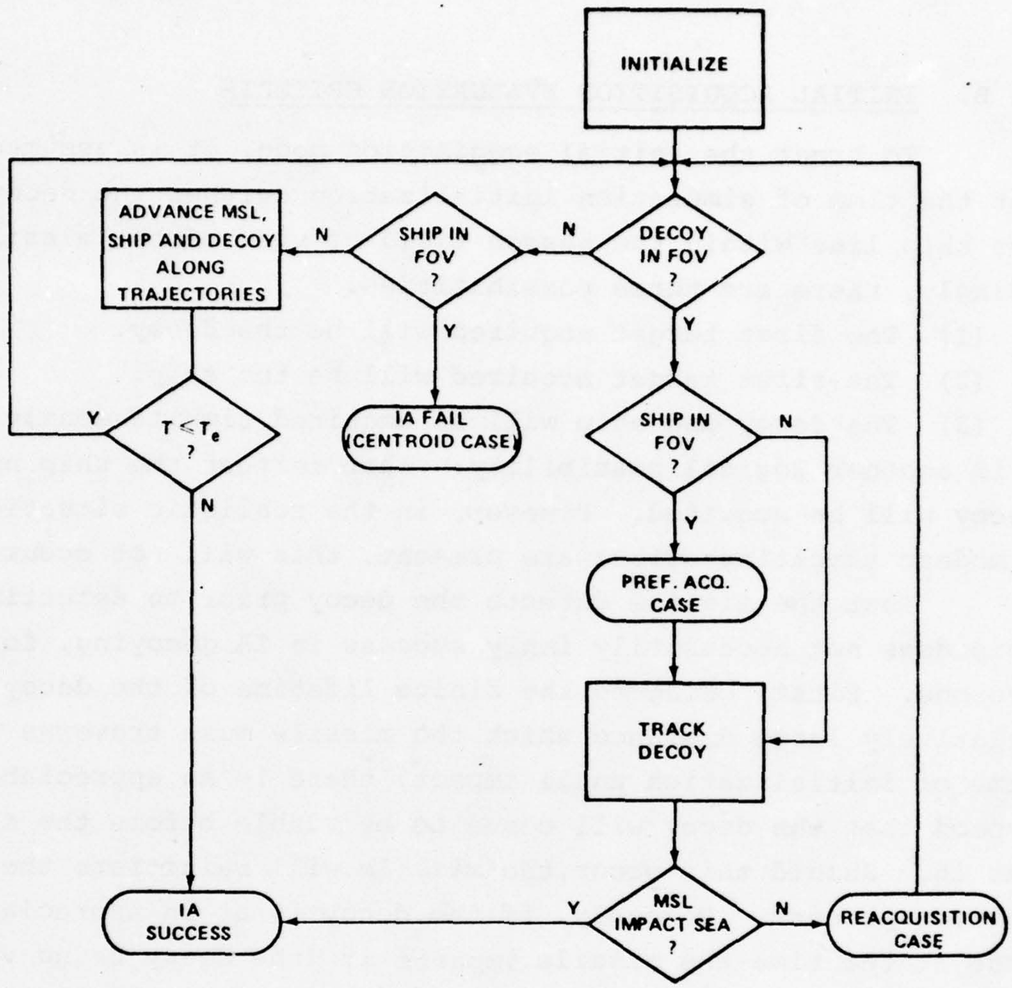


Figure 9

As indicated by Figure 9, a given engagement may have either of two outcomes: IA success or IA failure. Both may be described in terms of the additional qualifiers (preferred acquisition and reacquisition). IA success occurs if the missile fails to impact the ship for any reason, including failure of the missile to acquire any target whatsoever.

The missile utilized in IADS.ONE does not necessarily search for its target throughout its flight, but may initiate the process at some specified time. There are two such timing sequences built in to the simulation. According to the first, the search

process is undertaken at a specified time $\overline{T}$ prior to the predicted time of impact:

$$T_{acq}=T_{impact} -\overline{T}$$

The parameter T_{impact} is calculated based on the time of flight from initialization of the encounter, and includes the effects of targeting errors. From Figure 4 it is seen that the predicted time of impact, given the targeting errors ε and δR, will be

$$T_{impact} =(R_o+\delta R)/V_T \sin(\beta+\gamma-\varepsilon)$$

The other possibility is that the missile will commence its search at the instant the engagement begins: $T_{act}=0$. In an attempt to simulate this process while minimizing the cost of execution of the simulation, this mode of operation is accounted for by computing a value of $\overline{T}$ which is sufficiently large so that neither the decoy nor the ship can be within the missile field of view at the time of search initiation. (For the single set of runs for which data are available for both modes of seeker operation, the cost per event was approximately tripled in going from a specified $\overline{T}$ to the quasi-continuous search mode, while use of $T_{acq}=0$ would have increased the overall cost of execution by a factor of about five, for the particular set of parameters used.)

Given that $T>T_{acq}$, an object is considered to be within the missile field of view if

$$\alpha<\alpha_s$$

and

$$\delta_u<\delta<\delta_L \tag{26}$$

simultaneously, where α_s is the missile azimuth search FOV. The target azimuth angle is calculated using

$$\alpha=\mathrm{Tan}^{-1}(Y''/X'')$$

while the down look angle δ is

$$\delta=\mathrm{Tan}^{-1}(Z''/\sqrt{X''^2+Y''^2})$$

If at some time $T>T_{act}$ equations (26) hold for the ship and not for the decoy, an IA failure condition exists. In case the decoy or the decoy and the ship satisfy (26), the decoy is tracked. Then, if

(1) The decoy lifetime is sufficiently long that it remains viable until the missile reaches it and equation (18) is satisfied;

or

(2) The decoy loses viability prior to the missile's reaching it but the missile is unable to acquire the ship; or

(3) The missile flies through the decoy, resumes its search but is unable to acquire the ship

then an IA success condition exists. In the simulation cases (2) and (3) above are flagged as "Reacquisition Cases" regardless of whether or not the missile subsequently acquires the ship.

V. BREAK LOCK SIMULATION SOFTWARE

The Infrared Decoy Placement Simulation IRDPS.F4 was written in the version of the FORTRAN IV(G) language utilized by the DEC System 10 computer (DEC, 1972). The program was designed to facilitate its use with time-shared computer facilities, in particular those of TYMSHARE, Inc. Although FORTRAN IV(G) is a standard language and time sharing computing procedures and capabilities are fairly uniform, it is impossible to guarantee that the program described in this section can be utilized on other systems without alteration.

In order to execute IRDPS.F4, it is necessary to compile the program and create two input data files. The methods for accomplishing this using TYMSHARE facilities are described in detail in XEXEC (TYMSHARE, 1974) and EDITOR (TYMSHARE, 1969), so this section will concentrate on the file contents rather than the method for their creation.

The first of the input data files is referred to as the parameter file. It contains the parameters listed in Table 1; the layout is as shown in Table 2. In accordance with good programming style (Kernighan and Plauger, 1974) all parameter names consist entirely of alphabetical characters.

As indicated in Table 1, three seed numbers are required as input parameters; they are used in generating the true wind direction and speed, and the missile initial direction. Three seeds are required because, as shown by MacLaren and Marsaglia (1965) the multiplicative congruential generator cannot produce uniformly distributed triples. See also McQuay (1973). Since the seeds and the multiplier should be relatively prime (Newman and Odell, 1971), and also because the seeds should be relatively prime with respect to one another, all three seed numbers should be chosen to be prime. Further, since the period of the generator is maximized (and equal to 2^{b-2} where b is the word length of the computer) for ISEED=8N$\pm$3 where N is an integer (Newman and Odell, 1971), the seed numbers should all be of that form. Such is always possible, and can be accomplished with little effort (Carmichael, 1914).

NAME	EXPLANATION	RESTRICTIONS
IRPT	NUMBER OF CASES TO BE INCLUDED IN DETAILED REPORT	INTEGER ≤ IMMX
IC	VERSION IC = 0 GIVES MONTE CARLO RUN IC ≠ 0 GIVES DETERMINISTIC RUN	INTEGER
IMMX	NUMBER OF CASES IN RUN (IC = 0) NUMBER OF THREAT BEARINGS PER VALUE OF VW AND THETA (IC ≠ 0)	INTEGER ≤ 9999
ISEED (I)	SEEDS FOR RANDOM NUMBER GENERATORS FOR VW, THETA AND GAMMA, RESPECTIVELY	PRIME, OF FORM 8N ± 3 FOR SOME INTEGER N
VWL	MAXIMUM WIND SPEED, F/S (MONTE CARLO VERSION)	INTEGER ≤ 100
DELTAT	TIME INCREMENT BETWEEN SUCCESSIVE EVALUATIONS OF PLACEMENT, SECONDS	REAL > 0
GAMZRO	MINIMUM VALUE OF THREAT DIRECTION, DEG	0 ≤ GAMZRO ≤ GAMMAX
GAMMAX	MAXIMUM VALUE OF THREAT DIRECTION, DEG	GAMZRO ≤ GAMMAX ≤ 180°
TF	DECOY TIME OF FLIGHT, S	REAL > 0
TB	DECOY TIME OF SIGNATURE DEVELOPMENT, S	REAL ≥ 0
BSHD	HALF LENGTH OF SHIP, F	REAL > 0
RCH	HORIZONTAL RANGE OF DECOY, F	REAL > 0
DCCZ	HALF EXTENT OF DECOY IN A VERTICAL PLANE, F	REAL ≥ 0
HO	ALTITUDE OF DECOY AT TIME OF BURST, F	REAL ≥ 0
VCV	FALL RATE OF DECOY, F/S	REAL
DISTL	LOCATION OF DECOY REFERENCED TO AMIDSHIPS, F	REAL
VSKN	SHIP SPEED, KNOTS	INTEGER ≥ 0
IPHI	LAUNCHER TRAIN ANGLE REFERENCED TO SHIP LONGITUDINAL AXIS	0 ≤ INTEGER ≤ 180°
RZERO	MISSILE RANGE AT TIME OF DECOY LAUNCH, F	REAL > 0
ZTZERO	MISSILE INITIAL ALTITUDE, F	REAL > 0
VT	MISSILE SPEED, F/S	REAL > 0
GLIM	MISSILE AERODYNAMIC MANEUVER LIMIT, G'S	REAL > 0
ALPHO	MISSILE SINGLE DETECTOR AZIMUTH IFOV, DEG	REAL > 0
ALPHAS	MISSILE AZIMUTH SEARCH LIMIT, DEG	REAL > 0
SPAST	MISSILE AZIMUTH ANGLE OF SCAN PAST TARGET, DEG	REAL ≥ 0
DLTU	DEPRESSION ANGLE TO UPPER LIMIT OF MISSILE ELEVATION FOV, DEG	REAL > 0
DLTL	DEPRESSION ANGLE TO LOWER LIMIT OF ELEVATION FOV, DEG	REAL, DLTL > DLTU
ITS	NUMBER OF TIMES DECOY MUST BE IN FOV	INTEGER > 0
VWO	FOR IC ≠ 0, THE MINIMUM WIND SPEED, KN	REAL ≥ 0
VWI	FOR IC ≠ 0, THE WIND SPEED INCREMENT, KN	POSITIVE REAL SUCH THAT VWM = VWO + KVWI FOR SOME INTEGER K
VWM	FOR IC ≠ 0, THE MAXIMUM WIND SPEED, KN	REAL ≥ VWO
THETO	FOR IC ≠ 0, THE MINIMUM TRUE WIND DIRECTION, DEG	REAL, 0 ≤ THETO ≤ 360°
THETI	FOR IC ≠ 0, THE TRUE WIND DIRECTION INCREMENT, DEG	POSITIVE REAL SUCH THAT THETM + THETO + NTHETI FOR SOME INTEGER n
THETM	FOR IC ≠ 0, THE MAXIMUM TRUE WIND DIRECTION, DEG	REAL ≥ THETO

TABLE 1

The second of the input files contains the true wind speed distribution table, which is discussed in the Appendix. The table DIST contains 100 pairs of four digit integers (the program will accept 150 pairs without modification). The first entry is the true wind in feet per second; the second is a four digit number which is used in conjunction with the random number generator for the true wind speed to obtain a sample population resembling the desired parent population.

The program can be executed using DO loops to generate VW (true wind speed) and THETA (true wind direction) in addition to the Monte Carlo method of generation. This mode of operation can be of great utility when a limited range of wind conditions is of interest, as when evaluating decoy placement against a scenario which is specific in terms of geographic or weather conditions and ship direction of advance.

To run the program using TYMSHARE, the following sequence of operations is required. The sequence below assumes that the data files have been written and the program compiled. Underlined statements are entered by the user; the symbol ¢ indicates a carriage return.

```
(Log on to system)
- EXECUTE IRDPS.F4¢
LOADING
EXECUTION
ENTER PARAM FILE NAME:      PARAMF¢
ENTER NAME OF DIST TABLE:   DIST¢
ENTER SUM RPT FILE NAME:    SUMARY¢
ENTER DTL RPT FILE NAME:    DETAIL¢
(Program Executes)
TOTAL TRU FOR RUN=XXX
TRU PER CASE=YYY
EXIT
-(Log off from system or repeat sequence for another data set)
```

In the above example, the file PARAMF contained the parameters listed in Table 2, DIST was the table of true wind speed probabilities; the summary file SUMARY and the detail (case by case) file DETAIL were written. The last two lines gave the CPU use for the run and the CPU use per case.

If desired, the output files can be typed on the remote terminal simply by giving the command TYPE (File Name) ¢. Alternatively, they can be listed (at lower cost) on a remote high speed printer by means of the SPOOL command. Both files are width compatible with TTY size paper and half width terminals.

The remaining pages of this section contain complete listings for IRDPS.F4 and DIST. The main program requires approximately 15.3k bytes of core in the compiled version DIST requires an additional 0.9k bytes.

FIRST LINE: FORMAT (7I, 3F) "PROGRAM CONTROL"
IRPT, IMMX, IC, ISEED (1), ISEED (2), ISEED (3), VWL, DELTAT, GAMZRO, GAMMAX
SECOND LINE: FORMAT (8F, 2I) "DECOY SYSTEM/INSTALLATION AND SHIP PARAMETERS"
TF, TB, BSHD, RCH, DCCZ, HO, VCV, DISTL, VSKN, IPHI
THIRD LINE: FORMAT (9F, I) "MISSILE PARAMETERS"
RZERO, ZTZERTO, VT, GLIM, ALPHO, ALPHAS, SPAST, DLTU, DLTL, ITS
FOURTH LINE: FORMAT (10A4) "FILE NAME"
FILE NAME
FIFTH LINE: FORMAT (6F) "WIND PARAMETERS"
VWO, VWI, VWM, THETO, THETI, THETM

TABLE 2

```
      DOUBLE PRECISION NAME,DNAME,SNAME,FNAME
      INTEGER SUCS1,SUCS2,VWL,VSKN
      DIMENSION IVWT(2,150),ISEED(3),IC(12),IGF(2,12),
     -GS(2,12),IRW(24),IRWF(2,24),RWS(2,24),IWV(13),
     -IVWF(2,13),VWS(2,13),IGLO(12),IGHI(12),
     -IVLO(12),IVHI(12),IDLO(24),IDHI(24),IDSC(10)
      XSTRT=TRU(XXX)
C***  READ IN PARAM CARDS
      TYPE 4
    4 FORMAT(' ENTER PARAM FILE NAME: ',$)
      ACCEPT 7,FNAME
    7 FORMAT(A10)
      OPEN(8,FNAME,INPUT,SYMBOLIC,ERR=5)
      GO TO 10
    5 TYPE 6
    6 FORMAT('NO PARAM FILE--PROGRAM ABORTED.')
      STOP
   10 READ (8,11) IRPT,IMMX,IC,ISEED,VWL,DELTAT,GAMZRO,GAMMAX
   11 FORMAT(7I,3F)
      READ(8,12) TF,TB,BSHD,RCH,DCCZ,HO,VCV,
     -DISTL,VSKN,IPHI
   12 FORMAT(8F,2I)
      READ(8,13) RZERO,ZTZERO,VT,GLIM,ALPHO,ALPHAS,
     -SPAST,DLTU,DLTL,ITS
   13 FORMAT(9F,I)
      READ(8,14) (IDSC(I),I=1,10)
   14 FORMAT(10A4)
      READ(8,15)VWO,VWI,VWM,THETO,THETI,THETM
   15 FORMAT(6F)
      CLOSE(8)
C***  READ IN VW DISTRIBUTION TABLE.
      I=1
      TYPE 18
   18 FORMAT(' ENTER FILE NAME OF DIST TABLE: ',$)
      ACCEPT 7,NAME
      OPEN(9,NAME,INPUT,SYMBOLIC,ERR=21)
      GO TO 19
   21 TYPE 17
   17 FORMAT(' NO VW DISTRIBUTION TABLE - PROG. ABORTED.')
      STOP
   19 READ (9,16,END=20) IVWT(1,I),IVWT(2,I)
   16 FORMAT(2I4)
      I=I+1
      IF(I.GT.150) GO TO 20
      GO TO 19
   20 IVTMX=I-1
      CLOSE(9)
      DO 25 J=1,IVTMX
      IF(VWL.LE.IVWT(1,J)) GO TO 30
   25 CONTINUE
```

```
   30 IMAX=IVWT(2,J)
C*** INITILIZE COUNTERS
      ITHM=(THETM-THETO)/THETI+1.0
      IVWM=(VWM-VWO)/VWI+1.0
C*** OPEN I/O FILES.
      TYPE 37
   37 FORMAT(' ENTER SUM RPT FILE NAME: ',$)
      ACCEPT 7,SNAME
      OPEN(10,SNAME,OUTPUT,SYMBOLIC)
      IF(IRPT.GT.0) TYPE 38
   38 FORMAT(' ENTER DTL RPT FILE NAME: ',$)
      IF(IRPT.GT.0) ACCEPT 7,DNAME
      IF(IRPT.GT.0) OPEN(11,DNAME,OUTPUT,SYMBOLIC)
      G=32.17
      CRD=57.29578
      CNMF=6076
      VS=VSKN*CNMF/3600
      PHI=IPHI/CRD
      SPHI=SIN(PHI)
      CPHI=COS(PHI)
      SPAST=SPAST/CRD
      ALPHO=ALPHO/CRD
      ALPHAS=ALPHAS/CRD
      DLTL=DLTL/CRD
      DLTU=DLTU/CRD
      KAPPA=GLIM*G/(VT*ALPHAS)
C*****CLEAR ARRAYS
      DO 40 I=1,12
      IG(I)=0
      IWV(I)=0
   40 IWV(13)=0
      DO 45 I=1,24
      IRW(I)=0
   45 IFAIL1=0
      IFAIL2=0
      IFAIL=0
C*****LOOP THRU IM
   50 DO 2375 IM=1,IMMX
C*********************
C     IF DETERMINISTIC, LOOP THROUGH THETA AND VW
C*********************
      IF(IC.EQ.0) GO TO 60
      DO 2350 ITHET=1,ITHM
      THETA=THETO+(ITHET-1)*THETI
      THETA=THETA/CRD
      DO 2300 IVW=1,IVWM
      VW=VWO+(IVW-1)*VWI
      VW=VW*3600/CNMF
      STHETA=SIN(THETA)
      CTHETA=COS(THETA)
```

```
   60 CONTINUE
   76 IF(VWL.GT.0.0) GO TO 80
      GO TO 110
C*****GET RANDOM VW
      IF(IC.GT.0.0) GO TO 120
   80 YFL=RAN(ISEED(1))
      ITEMP=YFL*IMAX+.5
      IF(ITEMP.LT.1) GO TO 80
   85 DO 90 I=1,IVTMX
      IF(ITEMP.GT.IVWT(2,I)) GO TO 90
      GO TO 95
   90 CONTINUE
   95 IF(IVWT(1,I).GT.VWL) GO TO 80
      VW=IVWT(1,I)
      IVW=IVWT(1,I)
      IF(VW.EQ.0.0) GO TO 110
C*** GET RANDOM THETA.
  100 YFL=RAN(ISEED(2))
      K=YFL*379.0
      IF(K.LT.10.OR.K.GT.369) GO TO 100
      ITHETA=K-10
      THETA=ITHETA/CRD
      STHETA=SIN(THETA)
      CTHETA=COS(THETA)
      GO TO 120
  110 ITHETA=0
      THETA=0.0
      STHETA=0.0
      CTHETA=1.0
      VW=0.0
C*** GET RANDOM GAMMA.
  120 YFL=RAN(ISEED(3))
      K=YFL*(GAMMAX+19.0)
      IF(K.LT.(GAMZRO+10.0).OR.K.GT.(GAMMAX+10.0)) GO TO 120
      IGAMMA=K-10
      GAMMA=IGAMMA/CRD
      SGAMMA=SIN(GAMMA)
      CGAMMA=COS(GAMMA)
C*****CALCULATE TEND
      BETA=ASIN(VS*SGAMMA/VT)
      SBG=SIN(BETA+GAMMA)
      CBG=COS(BETA+GAMMA)
      IF(IGAMMA.NE.0.AND.IGAMMA.NE.180) GO TO 190
      IF(IGAMMA.EQ.180) TEND=RZERO/(VT-VS)
      IF(IGAMMA.EQ.0) TEND=RZERO/(VT+VS)
      GO TO 191
 190  TEND=RZERO*SGAMMA/(VT*SBG)
C*** CALCULATE RWA
  191 IRWA=0.0
      IF(VWL.EQ.0) GO TO 250
```

```
  200 TEMP=VW*CTHETA+VS
      IF(ITHETA.EQ.180) GO TO 210
      IF(TEMP.NE.0.0) GO TO 220
      IF(ITHETA.LT.180) GO TO 205
      ZETA=270.0
      GO TO 240
  205 ZETA=90.0
      GO TO 240
  210 IF(TEMP.LT.0.0) GO TO 215
      ZETA=0.0
      GO TO 240
  215 ZETA=180.0
      GO TO 240
  220 TEMP=(VW*STHETA)/TEMP
      IF(TEMP.LT.0.0) TEMP=-TEMP
      ZETA=ATAN(TEMP)
      ZETA=ZETA*CRD
      TEMP=VW*CTHETA
      IF(TEMP.LT.0.0) TEMP=-TEMP
      IF(ITHETA.GE.0.AND.ITHETA.LE.90) GO TO 240
      IF(ITHETA.LE.180) GO TO 230
      IF(ITHETA.LE.270) GO TO 235
  225 ZETA=360-ZETA
      GO TO 240
  230 IF(TEMP.LE.VS) GO TO 240
      ZETA=180-ZETA
      GO TO 240
  235 IF(TEMP.LE.VS) GO TO 225
      ZETA=ZETA+180
  240 IRWA=ZETA
C*** CALCULATE VRW
  250 VRW=SQRT(VW*VW+VS*VS+2*VW*VS*CTHETA)
C*** INITILIZE TIME.
  300 T=TB+TF
      P1=RCH*CPHI+DISTL+VS*TF
      P2=VW*CTHETA
      P3=RCH*SPHI
      P4=VW*STHETA
      P5=RZERO*SGAMMA
      P6=RZERO*CGAMMA
      X=(DLTL+DLTU)/2
      TGTX=SIN(X)/COS(X)
      ICNTS=0
      SUCS1=0
      SUCS2=0
      TSUCS1=0
C*****CALCULATE COORDS IN EARTH FRAME
  320 XC=P1-T*P2
      YC=P3-T*P4
      ZC=HO-VCV*(T-TF)
```

```
      IF(ZC.LT.0.0) ZC=0.0
      XS=VS*T
      XT=P6+VS*T-T*P6/TEND
  326 YT=P5*(1-T/TEND)
C*****CALCULATE COORDS IN TRACKER FRAME
      XDS=(XT-XS)*CBG+YT*SBG
      YDS=(XS-XT)*SBG+YT*CBG
      XDSB=(XT-XS-BSHD)*CBG+YT*SBG
      XDSS=(XT-XS+BSHD)*CBG+YT*SBG
      YDSB=(XS+BSHD-XT)*SBG+YT*CBG
      YDSS=(XS-BSHD-XT)*SBG+YT*CBG
      RHO=SQRT(XDS**2+YDS**2)
      IF(RHO.LE.ZTZERO/TGTX) GO TO 327
      ZT=ZTZERO
      GO TO 328
 327  ZT=RHO*TGTX
 328  ZDS=-ZT
      XDC=(XT-XC)*CBG+(YT-YC)*SBG
      YDC=(XC-XT)*SBG+(YT-YC)*CBG
      ZC1=ZC-DCCZ
      IF(ZC1.LT.0.0) ZC1=0.0
      ZC3=ZC+DCCZ
      IF(ZC3.LT.0.0) ZC3=0.0
      ZDC1=ZC1-ZT
      ZDC=ZC-ZT
      ZDC3=ZC3-ZT
C*** CALCULATE HCANG.
  600 AH=XDS**2+YDS**2
      BH=XDC*XDC+YDC*YDC
      CH=(XDS-XDC)*(XDS-XDC)+(YDS-YDC)*(YDS-YDC)
      Z3=(AH+BH-CH)/(2*SQRT(AH*BH))
      IF(Z3.GE.1.0) Z3=1.0
      HCANG=ACOS(Z3)
C*****CALCULATE SUBTENSE OF SHIP
  800 STUFF=(XDSS**2+YDSS**2+XDSB**2+YDSB**2-4*(BSHD**2)
     -)/(2*SQRT((XDSS**2+YDSS**2)*(XDSB**2+YDSB**2)))
      IF(STUFF.GT.1.0) STUFF=1.0
      PSI=ACOS(STUFF)
C*****EVALUATE SUCS1
      IF(PSI.LE.ALPHO) GO TO 801
      PSI=0.8*PSI
      IF(PSI.GT.(ALPHAS-SPAST)) GO TO 802
      ALPHA=ABS(PSI/2)+SPAST
      GO TO 803
 801  ALPHA=ALPHO/2+SPAST
      GO TO 803
 802  ALPHA=ALPHAS
 803  IF(ZDC.GE.0.0) GO TO 810
      ETA1=ABS(ATAN(ZDC1/SQRT(BH)))
      IF(ETA1.LE.DLTL.AND.ETA1.GE.DLTU) GO TO 804
```

```
      ETA2=ABS(ATAN(ZDC/SQRT(BH)))
      IF(ETA2.LE.DLTL.AND.ETA2.GE.DLTU) GO TO 804
      ETA3=ABS(ATAN(ZDC3/SQRT(BH)))
      IF(ETA3.GT.DLTL) GO TO 1000
      IF(ETA3.LT.DLTU) GO TO 810
  804 IF(ABS(HCANG).GT.ALPHA) GO TO 810
      ICNTS=ICNTS+1
      IF(ICNTS.GE.ITS) SUCS1=1
      IF(SUCS1.EQ.1) GO TO 890
 810  T=T+DELTAT
      IF(T.LE.TEND) GO TO 320
      GO TO 1000
C***  TRACK DECOY AND EVALUATE SUCS2
 890  TSUCS1=T
      ZMIN=(1-COS(X))*(VT*VT)/(GLIM*G)
     -+VT*DELTAT*SIN(X)
      ZTIP=ZC-VCV*DDIST/VT
      IF(ZTIP.LE.ZMIN) GO TO 940
      GAMPR=ATAN(YT/XT)
      IF(GAMPR.LT.0.0) GAMPR=GAMPR+180/CRD
      BETPR=GAMMA-GAMPR+BETA
  900 SGBP=SIN(GAMPR+BETPR)
      CGBP=COS(GAMPR+BETPR)
      XINT=XT-YT*SGBP/CGBP
      ARG=(XT**2+YT**2-XC*(XT-XINT)-YC*YT-XT*XINT)/
     -SQRT(((XC-XT)**2+(YC-YT)**2)*((XT-XINT)**2+YT**2))
      IF(ARG.GT.1.0) ARG=1.0
      ALPHD=ACOS(ARG)
      YDC=(XC-XT)*SGBP+(YT-YC)*CGBP
      IF(YDC.LT.0.0) ALPHD=-ALPHD
      BETPR=BETPR+KAPPA*ALPHD*DELTAT
      CGBP=COS(GAMPR+BETPR)
      SGBP=SIN(GAMPR+BETPR)
      XT=XT-VT*DELTAT*CGBP
      YT=YT-VT*DELTAT*SGBP
      XC=XC-P2*DELTAT
      YC=YC-P4*DELTAT
  910 XS=XS+VS*DELTAT
      XDS=(XT-XS)*CGBP+YT*SGBP
      XDC=(XT-XC)*CGBP+(YT-YC)*SGBP
      IF(XDS.LE.VT*DELTAT.OR.XDC.LE.VT*DELTAT) GO TO 920
      GAMPR=ATAN(YT/XT)
      IF(GAMPR.LT.0.0) GAMPR=GAMPR+180/CRD
      GO TO 900
  920 YDSB=(XS+BSHD-XT)*SGBP+YT*CGBP
      YDSS=YDSB-2*BSHD*SGBP
      IF(YDSB.GT.0.0.AND.YDSS.GT.0.0.OR.YDSB.LT.0.0.AND.
     -YDSS.LT.0.0) GO TO 940
      SUCS2=0
      GO TO 1000
```

```
  940 SUCS2=1
C*** WRITE DETAIL REPORT
 1000 CONTINUE
      IGAMMA=GAMMA*CRD
      IVRW=VRW*3600/CNMF+.5
      IF(IM.GT.IRPT) GO TO 2010
      IF(IM.GE.2) GO TO 1100
      WRITE(11,1050) FNAME,VSKN,IPHI
 1050 FORMAT(1X,A10,2X,'DETAILED REPORT FOR VS = ',I2,
     -'KNOTS AND PHI = ',I3,'DEGREES'/
     -4X,'GAMMA',5X,'RWA',5X,'VRW',2X,'SUCS1',2X,'TSUCS1',2X,'SUCS2'/
     -3X,'DEG.REL.  DEG.REL.  KN',10X,'SEC.'//)
 1100 WRITE(11,1150) IGAMMA,IRWA,IVRW,SUCS1,TSUCS1,SUCS2
 1150 FORMAT(5X,I3,6X,I3,5X,I3,3X,I1,5X,F4.1,5X,I1)
C***  ACCUMULATE RUN STATISTICS
 2010 IF(SUCS1.EQ.0) IFAIL1=IFAIL1+1
      IF(SUCS1.EQ.1.AND.SUCS2.EQ.0) IFAIL2=IFAIL2+1
      IF(SUCS1.EQ.0.OR.SUCS1.EQ.1.AND.SUCS2.EQ.0) IFAIL=IFAIL+1
C*** ACCUMULATE THREAT GEOMETRY AND REL. WIND ANGLE STATISTICS
      DO 2050 I=1,11
      M=15*(I-1)
      N=15*I
      IF(M.LE.IGAMMA.AND.N.GT.IGAMMA) IG(I)=IG(I)+1
      IF(M.LE.IGAMMA.AND.N.GT.IGAMMA.AND.SUCS1.EQ.0)
     -IGF(1,I)=IGF(1,I)+1
      IF(M.LE.IGAMMA.AND.N.GT.IGAMMA.AND.SUCS2.EQ.0.
     -AND.SUCS1.EQ.1) IGF(2,I)=IGF(2,I)+1
 2050 CONTINUE
      IF(165.LE.IGAMMA) IG(12)=IG(12)+1
      IF(165.LE.IGAMMA.AND.SUCS1.EQ.0)
     -IGF(1,12)=IGF(1,12)+1
      IF(165.LE.IGAMMA.AND.SUCS2.EQ.0.
     -AND.SUCS1.EQ.1) IGF(2,12)=IGF(2,12)+1
      DO 2100 I=1,24
      M=15*(I-1)
      N=15*I
      IF(M.LE.IRWA.AND.N.GT.IRWA) IRW(I)=IRW(I)+1
      IF(M.LE.IRWA.AND.N.GT.IRWA.AND.SUCS1.EQ.0)
     -IRWF(1,I)=IRWF(1,I)+1
      IF(M.LE.IRWA.AND.N.GT.IRWA.AND.SUCS2.EQ.0.
     -AND.SUCS1.EQ.1) IRWF(2,I)=IRWF(2,I)+1
 2100 CONTINUE
      DO 2200 I=1,12
      M=5*(I-1)
      N=5*I
      IF(M.LE.IVRW.AND.N.GT.IVRW) IWV(I)=IWV(I)+1
      IF(M.LE.IVRW.AND.N.GT.IVRW.AND.SUCS1.EQ.0)
     -IVWF(1,I)=IVWF(1,I)+1
      IF(M.LE.IVRW.AND.N.GT.IVRW.AND.SUCS2.EQ.0.
     -AND.SUCS1.EQ.1) IVWF(2,I)=IVWF(2,I)+1
```

```
 2200 CONTINUE
      IF(IVRW.LT.60) GO TO 2250
      IWV(13)=IWV(13)+1
      IF(SUCS1.EQ.0) IVWF(1,13)=IVWF(1,13)+1
      IF(SUCS2.EQ.0) IVWF(2,13)=IVWF(2,13)+1
 2250 IF(IC.EQ.0.0) GO TO 2375
C***   END VW LOOP
 2300 CONTINUE
C***   END THETA LOOP
 2350 CONTINUE
C*** END IM LOOP
 2375 CONTINUE
C*** WRITE SUMMARY REPORT
 2400 DO 2450 I=1,12
      IF(IG(I).NE.0) GO TO 2425
      GS(1,I)=0
      GS(2,I)=0
      GO TO 2450
 2425 GS(1,I)=FLOAT(IG(I)-IGF(1,I))/IG(I)
      GS(2,I)=FLOAT(IG(I)-IGF(2,I))/IG(I)
 2450 CONTINUE
      DO 2500 I=1,24
      IF(IRW(I).NE.0) GO TO 2475
      RWS(1,I)=0
      RWS(2,I)=0
      GO TO 2500
 2475 RWS(1,I)=FLOAT(IRW(I)-IRWF(1,I))/IRW(I)
      RWS(2,I)=FLOAT(IRW(I)-IRWF(2,I))/IRW(I)
 2500 CONTINUE
      DO 2600 I=1,13
      IF(IWV(I).NE.0) GO TO 2550
      VWS(1,I)=0
      VWS(2,I)=0
      GO TO 2600
 2550 VWS(1,I)=FLOAT(IWV(I)-IVWF(1,I))/IWV(I)
      VWS(2,I)=FLOAT(IWV(I)-IVWF(2,I))/IWV(I)
 2600 CONTINUE
      SR1=FLOAT(IMMX-IFAIL1)/IMMX
      SR2=FLOAT(IMMX-IFAIL2)/IMMX
      SR3=FLOAT(IMMX-IFAIL1-IFAIL2)/IMMX
      BSHD2=2*BSHD
      WRITE(10,2700) FNAME
 2700 FORMAT(15X,'INPUT FILE:',2X,A10)
      WRITE(10,2710) VSKN
 2710 FORMAT(15X,'SHIP SPEED:',4X,I2,1X,'KNOTS')
      WRITE(10,2720) IPHI
 2720 FORMAT(2X,'LAUNCHER POINTING ANGLE:',3X,I3,1X,'DEG REL')
      WRITE(10,2730) BSHD2
 2730 FORMAT(14X,'SHIP LENGTH:',3X,F4.0,1X,'FEET')
      WRITE(10,2740) DISTL
```

```
2740 FORMAT(3X,'LAUNCHER LOCATION:',2X,F5.1,1X,'FT FM CTR'/)
     WRITE(10,2750) IMMX
2750 FORMAT(2X,'RUN STATISTICS:  TOTAL NUMBER OF RUNS   :  ',I4)
     WRITE(10,2760) IFAIL1
2760 FORMAT(19X,'TOTAL NUMBER OF FAIL1 :   ',I4)
     WRITE(10,2770) SR1
2770 FORMAT(19X,'TYPE 1 SUCCESS RATIO  :   ',F4.2)
     WRITE(10,2780) IFAIL2
2780 FORMAT(19X,'TOTAL NUMBER OF FAIL2 :   ',I4)
     WRITE(10,2790) SR2
2790 FORMAT(19X,'TYPE 2 SUCCESS RATIO  :   ',F4.2)
     WRITE(10,2795) SR3
2795 FORMAT(19X,'CENTROID SUCCESS RATIO:   ',F4.2/)
     WRITE(10,2800)
2800 FORMAT(6X,'FAILURES BY TRACKER DIRECTION'/
    -2X,'DIRECTION',2X,'NO.OF',4X,'FAIL1',7X,'FAIL2'/
    -2X,'(DEG REL)',2X,'CASES',2X,'NO.',2X,'FRAC.',2X,'NO.',
    -2X,'FRAC.'//)
     DO 2820 I=1,12
     IGLO(I)=15*(I-1)
     IGHI(I)=15*I
     WRITE(10,2810) IGLO(I),IGHI(I),IG(I),IGF(1,I),GS(1,I),
    -IGF(2,I),GS(2,I)
2810 FORMAT(3X,I3,'-',I3,3X,I4,2X,I4,2X,F4.2,2X,I4,2X,F4.2)
2820 CONTINUE
     WRITE(10,2840)
2840 FORMAT(//3X,'FAILURES BY RELATIVE WIND DIRECTION'/
    -2X,'DIRECTION  NO.OF',4X,'FAIL1',7X,'FAIL2'/
    -2X,'(DEG REL)  CASES  NO.  FRAC.  NO.  FRAC.'//)
     DO 2860 I=1,24
     IDLO(I)=15*(I-1)
     IDHI(I)=15*I
     WRITE(10,2850) IDLO(I),IDHI(I),IRW(I),IRWF(1,I),RWS(1,I),
    -IRWF(2,I),RWS(2,I)
2850 FORMAT(3X,I3,'-',I3,3X,I4,2X,I4,2X,F4.2,2X,I4,2X,F4.2)
2860 CONTINUE
     WRITE(10,2870)
2870 FORMAT(//4X,'FAILURES BY RELATIVE WIND SPEED'/
    -3X,'SPEED  NO.OF',4X,'FAIL1',7X,'FAIL2'/
    -3X,'(KN)',3X,'CASES  NO.  FRAC.  NO.  FRAC.'//)
     DO 2890 I=1,12
     IVLO(I)=5*(I-1)
     IVHI(I)=5*I
     WRITE(10,2880) IVLO(I),IVHI(I),IWV(I),IVWF(1,I),VWS(1,I),
    -IVWF(2,I),VWS(2,I)
2880 FORMAT(3X,I2,'-',I2,2X,I4,2X,I4,2X,F4.2,2X,I4,2X,F4.2)
2890 CONTINUE
     WRITE(10,2900) IWV(13),IVWF(1,13),VWS(1,13),IVWF(2,13),VWS(2,13)
2900 FORMAT(3X,'GE 60',2X,I4,2X,I4,2X,F4.2,2X,I4,2X,F4.2/)
C######
```

```
      XNOW=TRU(XXX)-XSTRT
      TYPE 4006,XNOW
 4006 FORMAT(' TOTAL TRU FOR RUN = ',F8.4)
      XNO=XNOW/IMMX
      TYPE 4007,XNO
 4007 FORMAT(' TRU PER CASE = ',F10.7)
C#########
      STOP
      END
```

DIST PAGE ONE

```
 0   1
 1   4
 2  11
 3  27
 4  54
 5  97
 6 159
 7 243
 8 349
 9 479
10 632
11 808
121006
131224
141460
151712
161978
172255
182541
192834
203132
213432
223733
234033
244330
254623
264910
275191
285465
295729
305985
316231
326468
336695
346911
357117
367313
377498
387674
397840
407996
418143
428281
438411
448532
458645
468751
478850
488942
499027
```

DIST PAGE TWO

509106
519179
529247
539310
549368
559422
569472
579516
589560
599599
609635
619668
629698
639725
649750
659773
669794
679813
689831
699847
709862
719875
729887
739898
749908
759917
769925
779933
789940
799946
809952
819957
829962
839966
849970
859973
869976
879979
889982
899984
909986
919988
929990
939992
949993
959994
969995
979996
989997
999998

VI. INITIAL ACQUISITION SIMULATION SOFTWARE

The Initial Acquisition Decoy Simulation IADS.ONE is a FORTRAN IV (G) program written for use on a TYMSHARE-owned DEC System 10 computer. Like the BL simulation it was designed for use via a remote terminal.

While the BL simulation can probably be used with other machines without modification, IADS.ONE cannot, because the random number generation algorithm RANDU has been written into the program in IADS.ONE, while it is called up in IRDPS.F4. This was done for reasons of economy, as IADS.ONE may use as many as six random numbers per case, twice the number of IRDPS.F4. The particular subroutine in IADS.ONE is optimized for a 35 bit word length; it must be changed if a computer having a different word size is to be utilized.

Table III lists the input parameters required for execution of IADS.ONE. Table IV indicates the format of the parameter file. As with the BL simulation, IADS.ONE requires an additional input data file containing the table of probabilities for the true wind speed. An example of that table (which was utilized exclusively by SCI) is contained in Section V; it is discussed in the Appendix. The method for entering the files and for choosing seed numbers for the random number generators are as outlined in Section V.

Execution of the program IADS.ONE using a remote TYMSHARE terminal is accomplished in the following sequence of operator entered instructions. The symbology is as in Section V.

```
(Log on to System)
-EXECUTE IADS.ONE¢
LOADING
EXECUTION
ENTER PARAMETER FILE NAME:                 PARAMF¢
ENTER NAME OF WIND DISTRIBUTION TABLE:     DIST¢
ENTER NAME OF DETAILED REPORT:             DTLRPT¢
ENTER NAME OF SUMMARY REPORT:              SUMRPT¢
(Executes program)
EXIT
-(Log off from system or repeat sequence for another data set)
```

NAME	EXPLANATION	RESTRICTIONS
TL	TIME OF DECOY LAUNCH, S	INTEGER ≥ 0
TF	DECOY TIME OF FLIGHT, S	REAL ≥ 0
TB	DECOY TIME OF SIGNATURE DEVELOPMENT, S	REAL ≥ 0
TLIFE	DECOY LIFETIME, S	INTEGER ≥ 0
VCV	DECOY FALL RATE, F/S	REAL ≥ 0
IVWL	TRUE WIND LIMIT SPEED, F/S	INTEGER ≥ 0
IPHI	LAUNCHER TRAIN ANGLE REFERENCED TO SHIP LONGITUDINAL AXIS, DEG	$0 \leq$ INTEGER $\leq 180^{o}$
DISTL	LOCATION OF LAUNCHER REFERENCED TO AMIDSHIPS, F	REAL
RCH	HORIZONTAL RANGE OF DECOY, F	INTEGER > 0
ZCH	ALTITUDE OF DECOY AT TIME OF BURST, F	INTEGER ≥ 0
IBSHD	HALF OF SHIP LENGTH, F	INTEGER ≥ 0
IVSO	INITIAL SHIP SPEED, KN	INTEGER ≥ 0
IVSI	INCREMENT SHIP SPEED, KN	INTEGER ≥ 0
IVSM	MAXIMUM SHIP SPEED, KN	IVSM = IVSO + K $*$ IVSI FOR SOME INTEGER K
GAMZRO	MINIMUM VALUE OF THREAT DIRECTION, DEG	$0 \leq$ GAMZRO $\leq$ GAMMAX
GAMMAX	MAXIMUM VALUE OF THREAT DIRECTION, DEG	GAMZRO $\leq$ GAMMAX $< 180^{o}$
INAME	FILE NAME (A10)	10 CHARACTERS, MAXIMUM
TBAR	MISSILE SEEKER ACTIVATION TIME REFERENCED TO TIME OF PREDICTED IMPACT, S	REAL ≥ 0
RLPMAX	MAXIMUM INITIAL MISSILE RANGE, M	REAL ≥ 0
RLPMIN	MINIMUM INITIAL MISSILE RANGE, M	REAL ≥ 0
IRSD	RANGE TARGETING ERROR STANDARD DEVIATION, M	INTEGER ≥ 0
EPSD	ANGLE TARGETING ERROR STANDARD DEVIATION, DEG	REAL ≥ 0
VT	MISSILE SPEED, MACH	REAL > 0
ZTZERO	MISSILE INITIAL ALTITUDE, M	REAL > 0
ALPHAS	HALF OF TOTAL AZIMUTH SEARCH FOV, DEG	REAL > 0
DLTU	DEPRESSION ANGLE TO UPPER LIMIT OF MISSILE ELEVATION FOV, DEG	REAL > 0
DLTL	DEPRESSION ANGLE TO LOWER LIMIT OF MISSILE ELEVATION FOV, DEG	REAL $>$ DLTU
GLIM	MISSILE MANEUVER LIMIT, "G"	REAL > 0
DELTAT	TIME INCREMENT BETWEEN SUCCESSIVE EVALUATIONS, S	REAL > 0
IRPT	NUMBER OF CASES FOR WHICH DETAIL REPORT IS TO BE WRITTEN	INTEGER $\leq$ IM
JM	NUMBER OF CASES TO BE EXECUTED PER VALUE OF SHIP SPEED (VS)	INTEGER $\leq$ 9999
ISEED (I)	SEEDS FOR RANDOM NUMBER GENERATORS FOR GAMMA, THETA, VW, RLP, EPSLON, AND RERROR, RESPECTIVELY	PRIME, OF FORM $8N \pm 3$ FOR SOME INTEGER N

TABLE 3

FIRST LINE: FORMAT (3F, I, F, I) "TIMING, WIND AND DECOY"

TL, TF, TB, TLIFE, VCV, IVWL

SECOND LINE: FORMAT (I, F, 2I) "DECOY"

IPHI, DISTL, RCH, ZCH

THIRD LINE: FORMAT (4I, 2F) "SHIP PARAMETERS AND MISSILE DIRECTION"

IBSHD, IVSO, IVSI, IVSM, GAMZRO, GAMMAX

FOURTH LINE: FORMAT (A10) "FILE NAME"

FILE NAME

FIFTH LINE: FORMAT (3F, I, 8F) "MISSILE PARAMETERS"

TBAR, RLPMAX, RLPMIN, IRSD, EPSD, VT, ZTZERO, ALPHAS, DLTU, DLTL, GLIM, DELTAT

SIXTH LINE: FORMAT (8I) "PROGRAM CONTROL AND SEEDS"

IRPT, JM, ISEED (1), ISEED (2), ISEED (3), ISEED (4), ISEED (5), ISEED (6)

TABLE 4

In this example PARAMF was the parameteric file as indicated in Table 4 and DIST was the true wind speed distribution; both files must be created prior to typing in the EXECUTE command. Two output files - DTLRPT and SUMRPT - were created, and could be accessed prior to logoff using the TYPE command, as both are width compatible with half-width terminals and TTY paper.

The remainder of this section contains a listing of IADS.ONE. The true wind distribution table DIST which was used for the data runs under this contract is contained at the end of Section V. The compiled version of IADS.ONE requires approximately 12.3k bytes of core.

IADS.ONE   PAGE ONE

```
      DOUBLE PRECISION NAME,IDTL,ISUM,INAME,JNAME
      INTEGER TLIFE,RCH,ZCH,TL
      DIMENSION ISEED(6),IVWT(2,150)
C*****READ IN PARAMETER CARDS
      TYPE 4
    4 FORMAT(' ENTER PARAMETER FILE NAME:  ',$)
      ACCEPT 7,NAME
    7 FORMAT(A10)
      OPEN(8,NAME,INPUT,SYMBOLIC,ERR=5)
      GO TO 10
    5 TYPE 6
    6 FORMAT(' NO PARAMETER FILE - PROGRAM ABORTED')
      STOP
   10 READ(8,11) TL,TF,TB,TLIFE,VCV,IVWL
   11 FORMAT(3F,I,F,I)
      READ(8,12) IPHI,DISTL,RCH,ZCH
   12 FORMAT(I,F,2I)
   23 READ(8,24)IBSHD,IVSO,IVSI,IVSM,GAMZRO,GAMMAX
   24 FORMAT(4I,2F)
      READ(8,25) INAME
   25 FORMAT(A10)
      READ(8,26) TBAR,RLPMAX,RLPMIN,IRSD,EPSD,VT,ZTZRO,ALPHAS,DLTU,DLTL,
     -GLIM,DELTAT
   26 FORMAT(3F,I,8F)
      READ(8,27) IRPT,JM,ISEED(1),ISEED(2),ISEED(3),ISEED(4),ISEED(5),
     -ISEED(6)
   27 FORMAT(8I)
      CLOSE(8)
C*****READ IN VW DISTRIBUTION TABLE
      IF(IVWL.EQ.0) GO TO 37
      I=1
      TYPE 28
   28 FORMAT(' ENTER NAME OF WIND DISTRIBUTION TABLE:  ',$)
      ACCEPT 7,JNAME
      OPEN(9,JNAME,INPUT,SYMBOLIC,ERR=30)
      GO TO 32
   30 TYPE 31
   31 FORMAT(' NO VW DISTRIBUTION TABLE - PROGRAM ABORTED')
      STOP
   32 READ(9,33,END=34) IVWT(1,I),IVWT(2,I)
   33 FORMAT(2I4)
      I=I+1
      IF(I.GT.150) GO TO 34
      GO TO 32
   34 IVTMX=I-1
      CLOSE(9)
      DO 35 J=1,IVTMX
      IF(IVWL.LE.IVWT(1,J)) GO TO 36
   35 CONTINUE
   36 JMAX=IVWT(2,J)
```

```
      GO TO 38
   37 VW=0
      THETA=0
C*****CONVERT PARAMETERS TO MKS UNITS
   38 CMF=3.28
      CNMM=1852.471
      CRD=57.29578
      G=9.80
      PHI=IPHI/CRD
      GAMZRO=GAMZRO/CRD
      GAMMAX=GAMMAX/CRD
      DISTL=DISTL/CMF
      RCH=RCH/CMF
      ZCH=ZCH/CMF
      VCV=VCV/CMF
      BSHD=IBSHD/CMF
      VSO=IVSO*CNMM/3600
      VSM=IVSM*CNMM/3600
      IVSI=IVSI*CNMM/3600
      EPSD=EPSD/CRD
      VT=VT*331
      ZT=ZTZRO
      ALPHAS=ALPHAS/CRD
      DLTU=DLTU/CRD
      DLTL=DLTL/CRD
      VS=VSO
      KAPPA=GLIM*G/(VT*ALPHAS)
      TGTX=SIN((DLTU+DLTL)/2)/COS((DLTL+DLTU)/2)
C******INITIALIZE COUNTERS
      JCASE=1
      INUM=0
      IFLAG1=0
      IFLAG2=0
      IFLAG3=0
      IFLAG4=0
      IFLAG5=0
      IFLG1=0
      IFLG2=0
      IFLG3=0
      IFLG4=0
      IFLG5=0
      IFLG6=0
      IFLG7=0
      TACQ1=0.0
      TACQL=0.0
  100 TYPE 101
  101 FORMAT(' ENTER NAME OF DETAILED REPORT: ',$)
      ACCEPT 7,IDTL
      OPEN(10,IDTL,OUTPUT,SYMBOLIC)
      TYPE 102
```

```
  102 FORMAT(' ENTER NAME OF SUMMARY REPORT: ',$)
      ACCEPT 7,ISUM
      OPEN(11,ISUM,OUTPUT,SYMBOLIC)
C* GET A RANDOM GAMMA
  200 IY1=ISEED(1)*262147
      IF(IY1) 205,206,206
  205 IY1=IY1+34359738367+1
  206 YFL1=IY1
      YFL1=YFL1*.2910383046E-10
      ISEED(1)=IY1
  210 GAMMA=YFL1*3.14159
      IF(GAMMA.LT.GAMZRO.OR.GAMMA.GT.GAMMAX) GO TO 200
C* GET VW AND THETA
  220 IF(IVWL.EQ.0.0) GO TO 260
      IY2=ISEED(2)*262147
      IF(IY2) 225,226,226
  225 IY2=IY2+34359738367+1
  226 YFL2=IY2
      YFL2=YFL2*.2910383046E-10
      ISEED(2)=IY2
  230 THETA=YFL2*6.28319
  235 IY3=ISEED(3)*262147
      IF(IY3) 236,237,237
  236 IY3=IY3*34359738367+1
  237 YFL3=IY3
      YFL3=YFL3*.2910383046E-10
      ISEED(3)=IY3
      ITMP=YFL3*JMAX+.5
      IF(ITMP.LT.1) GO TO 235
  240 DO 245 I=1,IVTMX
      IF(ITMP.GT.IVWT(2,I)) GO TO 245
      GO TO 250
  245 CONTINUE
  250 IF(IVWT(1,I).GT.IVWL) GO TO 235
      VW=IVWT(1,I)/CMF
      IVW=IVWT(1,I)/CMF
      GO TO 270
  260 VW=0.0
      IVW=0
      THETA=0.0
      IRWA=0
      RS=VS
      GO TO 273
C*****CALCULATE RWA AND VRW
  270 ITHETA=THETA*CRD
      TEMP=VW*COS(THETA)+VS
      IF(ITHETA.EQ.180) GO TO 1210
      IF(TEMP.NE.0.0) GO TO 1220
      IF(ITHETA.LT.180) GO TO 1205
      ZETA=270.0
```

```
      GO TO 1240
 1205 ZETA=90.0
      GO TO 1240
 1210 IF(TEMP.LT.0.0) GO TO 1215
      ZETA=0.0
      GO TO 1240
 1215 ZETA=180.0
      GO TO 1240
 1220 TEMP=VW*SIN(THETA)/TEMP
      IF(TEMP.LT.0.0) TEMP=-TEMP
      ZETA=ATAN(TEMP)
      ZETA=ZETA*CRD
      TEMP=VW*COS(THETA)
      IF(TEMP.LT.0.0) TEMP=-TEMP
      IF(ITHETA.GE.0.AND.ITHETA.LE.90) GO TO 1240
      IF(ITHETA.LE.180) GO TO 1230
      IF(ITHETA.LE.270) GO TO 1235
 1225 ZETA=360.0-ZETA
      GO TO 1240
 1230 IF(TEMP.LE.VS) GO TO 1240
      ZETA=180.0-ZETA
      GO TO 1240
 1235 IF(TEMP.LE.VS) GO TO 1225
      ZETA=ZETA+180.0
 1240 IRWA=ZETA
      RS=SQRT(VW*VW+VS*VS+2*VW*VS*COS(THETA))
C*****GET RLP
  273 IF(RLPMAX.EQ.RLPMIN) GO TO 280
      DELR=RLPMAX-RLPMIN
  275 IY4=ISEED(4)*262147
      IF(IY4) 276,277,277
  276 IY4=IY4+34359738367+1
  277 YFL4=IY4
      YFL4=YFL4*.2910383046E-10
      ISEED(4)=IY4
      RLP=RLPMIN+YFL4*DELR
      GO TO 290
  280 RLP=RLPMAX
  290 BETA=ASIN(VS*SIN(GAMMA)/VT)
C*****GET EPSLON
      IF(EPSD.EQ.0.0) GO TO 320
  300 A=0.0
      DO 310 I=1,12
  305 IY5=ISEED(5)*262147
      IF(IY5) 306,307,307
  306 IY5=IY5+34359738367+1
  307 YFL5=IY5
      YFL5=YFL5*.2910383046E-10
      ISEED(5)=IY5
  310 A=A+YFL5
```

```
      EPSLON=(A-6.0)*EPSD
      BETA=BETA+EPSLON
  320 TEND=RLP*SIN(GAMMA)/(VT*SIN(BETA+GAMMA))
C*****GET RERROR
      IF(IRSD.EQ.0.0) GO TO 350
  330 A=0.0
      DO 340 I=1,12
  335 IY6=ISEED(6)*262147
      IF(IY6) 336,337,337
  336 IY6=IY6+34359738367+1
  337 YFL6=IY6
      YFL6=YFL6*.2910383046E-10
      ISEED(6)=IY6
  340 A=A+YFL6
      RERROR=(A-6.0)*IRSD
      GO TO 360
  350 RERROR=0.0
  360 IF(TBAR.NE.0.0) GO TO 365
      COTDU=COS(DLTU)/SIN(DLTU)
      TBAR=(ZTZRO*COTDU+VS*TLIFE-RCH*COS(PHI)-DISTL+VT*DELTAT)/VT
     -+TL+1.0
  365 TACT=(RERROR+RLP*SIN(GAMMA))/(VT*SIN(BETA+GAMMA-EPSLON))-TBAR
      IF(TACT.LT.0.0) TACT=0.0
      T=TACT
C*****CALCULATE DECOY COORDS IN EARTH FRAME
  370 IF(T.GT.TL+TF+TB+TLIFE) GO TO 390
      XD=VS*(TL+TF)+DISTL+RCH*COS(PHI)
     --VW*COS(THETA)*(T-TL)
      YD=RCH*SIN(PHI)-VW*SIN(THETA)*(T-TL)
      ZD=ZCH-VCV*(T-TL-TF)
      IF(ZD.LT.0.0) ZD=0.0
C*****CALCULATE DECOY COORDS IN MOVING FRAME
      XDPRD=(RLP*COS(GAMMA)-XD)*COS(BETA+GAMMA)
     -+(RLP*SIN(GAMMA)-YD)*SIN(BETA+GAMMA)-VT*T
      YDPRD=(XD-RLP*COS(GAMMA))*SIN(BETA+GAMMA)
     -+(RLP*SIN(GAMMA)-YD)*COS(BETA+GAMMA)
      ZDPRD=ZD-ZT
C*****CALCULATE SHIP COORDS IN MOVING FRAME
  390 XDPRS=(RLP*COS(GAMMA)-VS*T)*COS(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*SIN(BETA+GAMMA)-VT*T
      XDSS=(RLP*COS(GAMMA)-BSHD-VS*T)*COS(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*SIN(BETA+GAMMA)-VT*T
      XDSB=(RLP*COS(GAMMA)+BSHD-VS*T)*COS(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*SIN(BETA+GAMMA)-VT*T
      YDPRS=(VS*T-RLP*COS(GAMMA))*SIN(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*COS(BETA+GAMMA)
      YDSB=(VS*T-BSHD-RLP*COS(GAMMA))*SIN(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*COS(BETA+GAMMA)
      YDSS=(VS*T+BSHD-RLP*COS(GAMMA))*SIN(BETA+GAMMA)
     -+RLP*SIN(GAMMA)*COS(BETA+GAMMA)
```

```
  400 ICOUNT=0
C*****DETERMINE IF SHIP IN FOV
  410 IF(XDPRS.GT.0.0) GO TO 415
      GO TO 430
  415 IF(YDSB.GE.0.0.AND.YDSS.LE.0.0.OR.YDSB.LE.0.0.AND.YDSS.GE.0.0)
     -ALPHS=0.0
      IF(XDSB.EQ.0.0.OR.XDSS.EQ.0.0) ALPHS=0.0
      IF(ALPHS.EQ.0.0)GO TO 420
      BAZANG=ABS(ATAN(YDSB/XDSB))
      SAZANG=ABS(ATAN(YDSS/XDSS))
      IF(BAZANG.GT.SAZANG) GO TO 417
      ALPHS=BAZANG
      GO TO 420
  417 ALPHS=SAZANG
  420 IF(ABS(ALPHS).GT.ALPHAS) GO TO 430
      DSHP=SQRT(XDPRS**2+YDPRS**2)
      ETAS=ABS(ATAN(ZT/DSHP))
      IF(ABS(ETAS).LE.DLTL.AND.ABS(ETAS).GE.DLTU) GO TO 500
C*****SHIP NOT IN FOV
  430 IF(T.GT.TL+TB+TF+TLIFE) GO TO 460
      IF(T.LT.TL+TF+TB) GO TO 460
      IF(XDPRD.GT.0.0) GO TO 440
      GO TO 460
  440 ALPHD=ATAN(YDPRD/XDPRD)
      IF(ABS(ALPHD).LE.ALPHAS) GO TO 445
      GO TO 460
  445 ETAD=ATAN(ZDPRD/(SQRT(XDPRD**2+YDPRD**2)))
      IF(ABS(ETAD).GE.DLTU.AND.ABS(ETAD).LE.DLTL) GO TO 450
      GO TO 460
  450 ICOUNT=ICOUNT+1
      TACQ=T
      IF(TACQ1.GT.0.0) TACQL=T
      IF(TACQ1.EQ.0.0) TACQ1=T
      GO TO 605
  460 T=T+DELTAT
      IF(T.GT.TEND) GO TO 700
      GO TO 370
C*****SHIP IN FOV
  500 IF(T.GT.TB+TL+TF+TLIFE) GO TO 515
      GO TO 520
  515 TACQ=T
      TACQ1=T
      IFLAG1=1
      IFLG1=IFLG1+1
      GO TO 700
  520 IF(T.LT.TL+TB+TF) GO TO 515
  521 XD=VS*(TL+TF)+DISTL+RCH*COS(PHI)
     --VW*COS(THETA)*(T-TL-TF)
      YD=RCH*SIN(PHI)-VW*SIN(THETA)*(T-TL-TF)
      ZD=ZCH-VCV*(T-TL-TF)
```

```
      IF(ZD.LT.0.0) ZD=0.0
      XDPRD=(RLP*COS(GAMMA)-XD)*COS(BETA+GAMMA)
     -+(RLP*SIN(GAMMA)-YD)*SIN(BETA+GAMMA)-VT*T
      YDPRD=(XD-RLP*COS(GAMMA))*SIN(BETA+GAMMA)
     -+(RLP*SIN(GAMMA)-YD)*COS(BETA+GAMMA)
      IF(XDPRD.GT.0.0) GO TO 530
      GO TO 515
  530 ALPHD=ATAN(YDPRD/XDPRD)
      IF(ABS(ALPHD).LE.ALPHAS.AND.ZT.GE.ZD) GO TO 540
      GO TO 515
  540 ETAD=ABS(ATAN(ZDPRD/(SQRT(XDPRD**2+YDPRD**2))))
      IF(ETAD.GE.DLTU.AND.ETAD.LE.DLTL) GO TO 550
      GO TO 515
  550 ICOUNT=ICOUNT+1
      TACQ=T
      IF(TACQ1.GT.0.0) TACQL=T
      IF(TACQ1.EQ.0.0) TACQ1=T
  560 IFLAG2=IFLAG2+1
      IF(IFLAG2.EQ.1) IFLG2=IFLG2+1
      GO TO 605
  605 TMPACT=SQRT(XDPRD**2+YDPRD**2)/VT
      IF(TMPACT+TACQ.GT.TL+TB+TF+TLIFE) GO TO 620
      ZMIN=(1-COS((DLTL+DLTU)/2))*VT**2/(G*GLIM)
     -+VT*DELTAT*SIN((DLTL+DLTU)/2)
      ZCTIP=ZD-VCV*(TMPACT-TL-TF)
      IF(ZCTIP.LT.0.0) ZCTIF=0.0
      IF(ZCTIP.GT.ZMIN) GO TO 620
  610 IFLAG4=1
      IFLG4=IFLG4+1
      GO TO 700
  620 IFLAG5=IFLAG5+1
      IF(IFLAG5.EQ.1) IFLG5=IFLG5+1
      ALPHA=ALPHD
      IF(T.GE.TEND) GO TO 660
C*****TRACK DECOY
  630 XT=RLP*COS(GAMMA)-VT*T*COS(BETA+GAMMA)
      YT=RLP*SIN(GAMMA)-VT*T*SIN(BETA+GAMMA)
      GAMPR=ATAN(YT/XT)
      IF(GAMPR.LT.0.0)GAMPR=GAMPR+180/CRD
      BETPR=GAMMA-GAMPR+BETA
  640 SGBP=SIN(GAMPR+BETPR)
      CGBP=COS(GAMPR+BETPR)
      XINT=XT-YT*SGBP/CGBP
      ARG=(XT**2+YT**2-XD*(XT-XINT)-YD*YT-XT*XINT)/
     -SQRT(((XD-XT)**2+(YD-YT)**2)*((XT-XINT)**2+TY**2))
      IF(ARG.GT.1.0) ARG=1.0
      ALPHD=ACOS(ARG)
      YDD=(XD-XT)*SGBP+(YT-YD)*CGBP
      IF(YDD.LT.0.0) ALPHD=-ALPHD
      BETPR=BETPR+KAPPA*ALPHD*DELTAT
```

```
      CGBP=COS(GAMPR+BETPR)
      SGBP=SIN(GAMPR+BETPR)
      XT=XT-VT*DELTAT*CGBP
      YT=YT-VT*DELTAT*SGBP
      XD=VS*(TL+TF)+DISTL+RCH*COS(PHI)
     --VW*COS(THETA)*(T-TL)
      YD=RCH*SIN(PHI)-VW*SIN(THETA)*(T-TL)
      ZD=ZD-VCV*DELTAT
      IF(ZD.LT.0.0) ZD=0.0
      RHOD=SQRT((XT-XD)**2+(YT-YD)**2)
      IF(RHOD.GE.(ZT-ZD)/TGTX) GO TO 650
      ZT=ZD+RHOD*TGTX
  650 XS=XS+VS*DELTAT
      XDS=(XT-XS)*CGBP+YT*SGBP
      GAMPR=ATAN(YT/XT)
      IF(GAMPR.LT.0.0) GAMPR=GAMPR+180/CRD
      IF(XDS.LE.0.0) GO TO 660
      T=T+DELTAT
      IF(T.LT.TL+TB+TF+TLIFE) GO TO 640
      ETAS=ABS(ATAN(ZT/SQRT((XT-XS)**2+YT**2)))
      IF(ETAS.GE.DLTL) GO TO 660
      GO TO 390
  660 IFLAG4=1
      IFLG4=IFLG4+1
C*****WRITE DETAILED REPORT
  700 IF(IFLAG2.GT.1) IFLG6=IFLG6+1
      IF(IFLAG5.GT.1) IFLG7=IFLG7+1
      INUM=INUM+1
      IF(INUM.GT.IRPT) GO TO 703
      IRERОR=RERROR
      RLPK=RLP/1000.
      VSKN=VS*3600/CNMM
      IGAM=GAMMA*CRD
      IRS=RS*3600/CNMM+.5
      EPSS=EPSLON*1000
      IF(INUM.GT.1) GO TO 704
      WRITE(10,701) NAME,VSKN
  701 FORMAT(1X,A10,12X,'DETAILED REPORT FOR VS = ',F3.0,'KNOTS'/
     -4X,'RLP  GAMMA TGTING ERROR TACQ. SEC CENT PREF MULT REACQ REL.',
     -    'WIND'/
     -4X,'KM',4X,'DEG  ANGLE  RANGE  1ST  LST ROID  ACQ  ACQ',7X,
     -    'SPD  ANG'/)
  704 WRITE(10,702) RLPK,IGAM,EPSLON,IREROR,TACQ1,TACQL,IFLAG1,IFLAG2,
     -               IFLAG3,IFLAG5,IRS,IRWA
  702 FORMAT(3X,F5.1,2X,I3,2X,F6.3,1X,I5,1X,F4.1,1X,F4.1,2X,3(I1,4X),I1,
     -3X,I2,3X,I3)
  703 JCASE=JCASE+1
      IF(JCASE.GT.JM) GO TO 800
      IFLAG1=0
      IFLAG2=0
```

```
      IFLAG3=0
      IFLAG4=0
      IFLAG5=0
      TACQ1=0
      TACQL=0
      ZT=ZTZRO/CMF
      GO TO 200
C*****WRITE SUMMARY REPORT
  800 BSHD2=2*BSHD
      TM=JM
      RF1=IFLG1/TM
      RF2=IFLG2/TM
      RF3=IFLG3/TM
      RF4=IFLG4/TM
      RF5=IFLG5/TM
      RF6=IFLG6/TM
      RF7=IFLG7/TM
      VSKN=VS*3600/CNMM
      RLPLUS=RLPMAX/1000
      RMINUS=RLPMIN/1000
      WRITE(11,805) VSKN,INAME,JM
  805 FORMAT(40X,'IADS SUMMARY REPORT'//,
     -42X,'VS',F3.0,' KNOTS'//,
     -1X,'***** THREAT DATA ******',13X,9(1H*),1X,'RUN DATA',14(1H*)/,
     -1X,'THREAT NAME: ',A10,15X,'TOTAL NO. OF CASES  ',I5)
      WRITE(11,806) RLPLUS,RMINUS,IFLG1,EPSS,RF1,RSD
  806 FORMAT(6X,'RLPMAX: ',F5.1,' KM'/,
     -6X,'RLPMIN: ',F5.1,' KM',17X,'NO. OF CENTROID CASES ',I4/,
     -8X,'EPSD: ',F4.1,' MRAD',32X,'RATIO ',F4.2/,
     -9X,'RSD: ',F4.2,' KM')
      WRITE(11,807) IFLG2,TLIFE,RF2,TF,TB,IFLG6,VCV,RF6
  807 FORMAT(1X,'*** DECOY INVARIANTS ***',14X,
     -'NO. OF PREF. ACQUISITIONS ',I4/,
     -7X,'TLIFE:',2X,I3,1X,'SEC.',36X,'RATIO ',F4.2/
     -10X,'TF: ',F4.1,' SEC.',/,
     -10X,'TB: ',F4.1,' SEC.',16X,'NO. OF MULT PA S  ',I4,/,
     -9X,'VCV: ',F4.1,' M/SEC.',26X,'RATIO ',F4.2//)
      WRITE(11,808) IFLG3,BSHD2,RF3,IVWL,DELTAT,IFLG4,RF4
  808 FORMAT(39X,'NO. OF MULT.DECOY ACQ. ',I4,/,
     -1X,'SHIP LENGTH: ',F5.1,' METERS',31X,'RATIO ',F4.2/,
     -9X,'VWL: ',I4,'FT/SEC',/,
     -6X,'DELTAT: ',F4.2,' SEC',17X,'NO. OF SUCCESSFUL IA. S ',I4,/,
     -57X,'RATIO ',F4.2,//)
      WRITE(11,809) IFLG5,RF5,IFLG7,RF7
  809 FORMAT(39X,'NO.OF REACQUISITIONS ',I4/,
     -53X,'RATIO ',F4.2//,
     -39X,'NO. OF MULT. REACQUISITIONS ',I4/,
     -53X,'RATIO ',F4.2///)
      WRITE(11,810)
  810 FORMAT(1X,'DECOY NO',3X,'1',5X,'2',5X,'3',5X,'4',5X,'5',5X,'6',
```

```
     -5X,'7',5X,'8',5X,'9',5X,'10',4X,'11',4X,'12')
      WRITE(11,811) IPHI
811   FORMAT(6X,'PHI',I3)
      WRITE(11,812) DISTL
812   FORMAT(4X,'DISTL ',F4.1)
      WRITE(11,813) RCH
813   FORMAT(6X,'RCH ',I3)
      WRITE(11,814) ZCH
814   FORMAT(6X,'ZCH ',I3)
      WRITE(11,815) TL
815   FORMAT(2X,'TLAUNCH ',I2)
      VS=VS+IVSI
      IF(VS.GT.VSM.OR.IVSI.EQ.0.0) GO TO 820
      JCASE=1
      IFLG1=0
      IFLG2=0
      IFLG3=0
      IFLG4=0
      IFLG5=0
      IFLG6=0
      IFLG7=0
      IFLAG1=0
      IFLAG2=0
      IFLAG3=0
      IFLAG4=0
      IFLAG5=0
      INUM=0
      TACQ1=0
      TACQL=0
      ZT=ZTZRO
      GO TO 200
820   CONTINUE
      END
```

VII. REFERENCES

Long, M.W., R.D. Wetherington, J.L. Edwards and A.B. Abeling, "Wavelength Dependence of Sea Echo," Engineering Experiment Station, Georgia Institute of Technology, Final Report, Contract N62269-3019, July 1965 (AD 477 905)

Schiff, S.C., and M.K. Pieklo, "Ship Launched Airborne Infrared Decoy Placement Study (U)", Systems Consultants, Inc., Final Report, Contract N60921-75-C-0237, February 1976 (AD C 005 120)

Digital Equipment Corporation, "DEC System 10 FORTRAN IV Programmer's Reference Manual," DEC-10-AFDO-D, May 1972

TYMSHARE, Inc., "XEXEC", TYMSHARE Manual TYMCOM-X, December 1974

TYMSHARE, Inc., "EDITOR", TYMSHARE Reference Manual, July 1969

Kernighan, B.W., and P.J. Plauger, "The Elements of Programming Style", McGraw-Hill, 1974

MacLaren, M.D., and G. Marsaglia, "Uniform Random Number Generators", J.ACM, Vol. 12 No. 1, January 1965

McQuay, W.K., "Computer Simulation Methods for Military Operations Research", Air Force Avionics Laboratory Technical Report AFAL-TR-73-341, October 1973 (AD 771 614)

Newman, T.G., and P.L. Odell, "The Generation of Random Variates", Griffin, 1971

Carmichael, R.D., "The Theory of Numbers and Diophantine Analysis", Dover, 1914

APPENDIX

THE DISTRIBUTION OF TRUE WIND SPEEDS AT SEA AND AN ALGORITHM FOR CALCULATING RELATIVE WIND

A.1 DISTRIBUTION OF TRUE WIND SPEEDS

The decoy placement simulations described in this volume require input data concerning true wind conditions since these, along with the ship's speed, determine the relative motion of a free falling airborne decoy with respect to the ship, and hence also with respect to the missile. Whereas the direction of the true wind can be assumed to be uniformly distributed to good approximation in modeling ensembles of encounters, the true wind speed cannot. Hence, it was desirable to obtain empirical data on true wind speed frequencies of occurrence in a maritime environment.

Fortunately, such data were reported by Long and his co-workers (Long, et al, 1965). Their report contained a curve giving true wind speeds and their frequencies of occurrence; the curve resulted from the aggregation of data for a large number of locations in several oceans over the period of many years.

Long's curve was analyzed to determine if it was reasonably well approximated by any known probability density function. The results of the analysis indicated an excellent fit for a slightly modified χ^2 distribution having eight degrees of freedom. The words "slightly modified" are significant here, even though the modification was extremely slight*. It must be emphasized that no claim is being made for a χ^2 distribution with 8 degrees of freedom being the result of some underlying property of nature. We could have done just as well with a polynomial curve fit.

Once a satisfactory fit to Long's curve was obtained, it was quantized. The result, plotted again in the form of a curve, is shown in Figure A1. Arranged in tabular form, the curve yields the table DIST.

A.2 CALCULATION OF RELATIVE WIND SPEED AND ANGLE

Given the direction of the wind θ as measured in the earth

*It consisted of assigning a non-zero probability ($P=10^{-4}$) to VW=0

PRECEDING PAGE BLANK-NOT FILMED

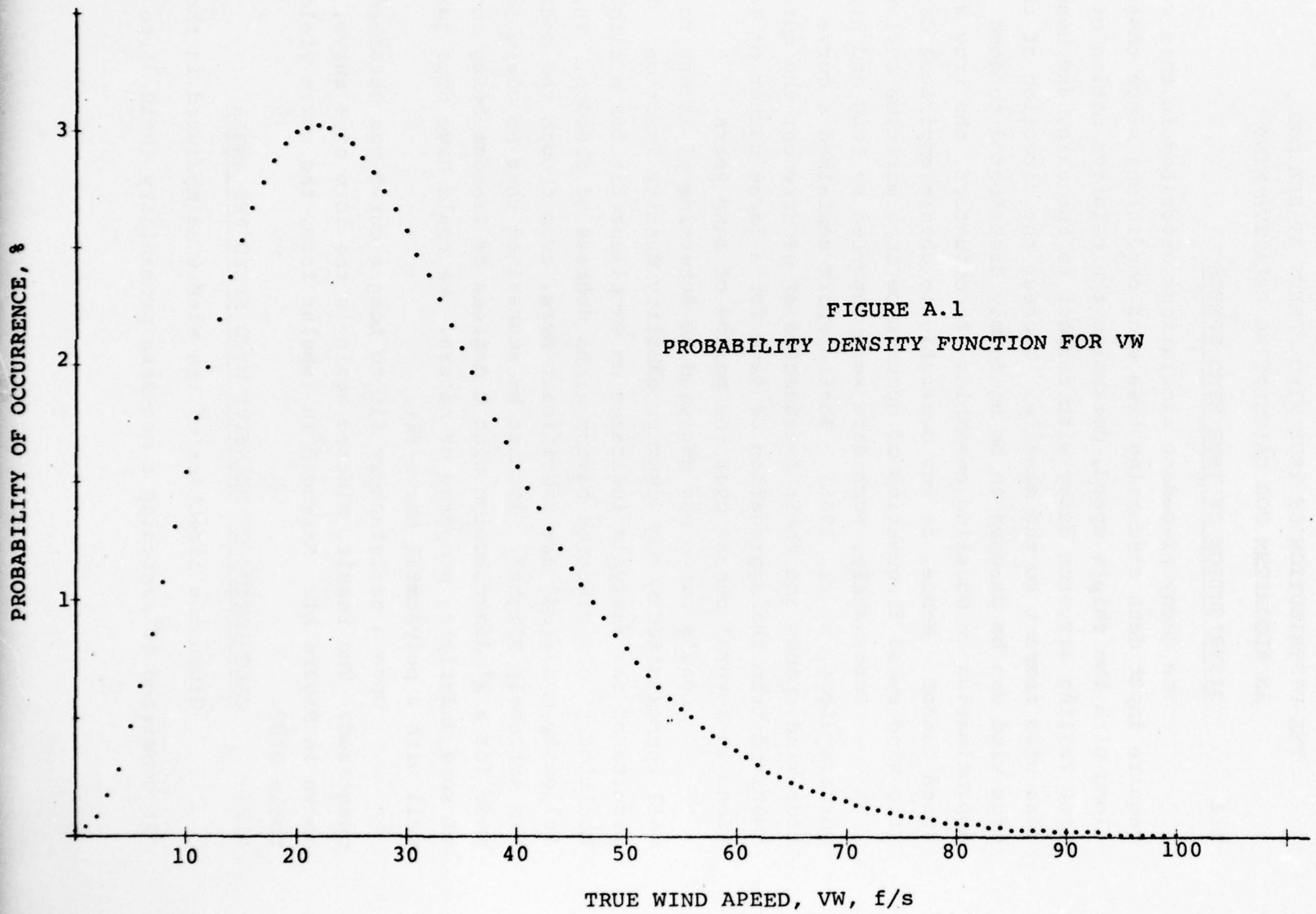


FIGURE A.1
PROBABILITY DENSITY FUNCTION FOR VW

frame, the wind speed V_W and the ship speed V_S, the relative wind speed V_{RW} and the relative wind direction β can be determined by solving the vector diagram shown in Figure A.1. Using the law of cosines,

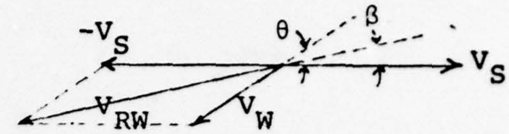


Figure A.1

$$V_{RW} = (V_S^2 + V_W^2 + 2V_W V_S \cos\theta)^{\frac{1}{2}} \qquad \text{A.1}$$

Using the law of sines,

$$\beta = \sin^{-1}(V_W \sin\theta / V_{RW}) \qquad \text{A.2}$$

Now equation (A.1) uniquely specifies V_{RW} for any triple (V_W, V_S, θ); however (A.2) is multivalued: for given V_W, V_{RW} and $\theta \neq N+1)\pi/2$, it admits roots β_1 and β_2 such that $\beta_2 = \pi - \beta_1$. Clearly, only one of the roots can be physically meaningful, but ostensibly it could be either one. For example, if $\theta=0$, β will be equal to 0° if $V_S > V_W$, and equal to 180° if $V_W > V_S$. The generalization of this observation, along with recognition of the fact that β will always be in the hemicircle containing θ, V_S and $-V_S$ allows design of the algorithm for calculating the relative wind angle. The rules for calculation of β are as follows:

1) If θ lies in the first quadrant, so does
2) If θ lies in the second quadrant, β lies in the second quadrant if
   $|V_W \cos\theta| > V_S$;
   otherwise it lies in the first quadrant
3) If θ lies in the third quadrant, β is also in the third quadrant if
   $|V_W \cos\theta| > V_S$;
   otherwise β lies in the fourth quadrant
4) If θ lies in the fourth quadrant, β is also in the fourth quadrant.